UNIVERZA V LJUBLJANI

FAKULTETA ZA RAČUNALNIŠTVO IN INFORMATIKO

Sašo Sotlar

Sledenje uporabnikom mobilnih naprav

DIPLOMSKO DELO

VISOKOŠOLSKI STROKOVNI ŠTUDIJSKI PROGRAM PRVE STOPNJE RAČUNALNIŠTVO IN INFORMATIKA

Mentor: dr. Dejan Lavbič

Ljubljana, 2013

Univerza *v Ljubljani*
Fakulteta *za računalništvo in informatiko*

Tržaška 25
1001 Ljubljana, Slovenija
telefon: 01 476 84 11
01 476 83 87
faks: 01 426 46 47
01 476 87 11
www.fri.uni-lj.si
e-mail: dekanat@fri.uni-lj.si

Št. naloge: 00386 / 2013
Datum: 4.4.2013

Univerza v Ljubljani, Fakulteta za računalništvo in informatiko izdaja naslednjo nalogo:

Kandidat: **SAŠO SOTLAR**

Naslov: **SLEDENJE UPORABNIKOM MOBILNIH NAPRAV**

TRACKING OF MOBILE DEVICE USERS

Vrsta naloge: Diplomsko delo visokošolskega strokovnega študija prve stopnje

Tematika naloge:

Brezična omrežja so na voljo na vedno več lokacijah, kjer se zbira večje število ljudi. Uporabniki mobilnih naprav, predvsem pametnih telefonov, želijo biti na vsakem koraku povezani, zato se v težnji po prostem dostopu do spleta pogosto poslužujejo takšnih brezžičnih omrežij. Številni uporabniki se sploh ne zavedajo dejstva, da njihov mobilni telefon z vključenim brezžičnim omrežjem samodejno oddaja določene podatke o napravi in uporabniku ter na ta način nezavedno puščajo svojo sled. V okviru diplomske naloge naj študent izdela prototip, ki omenjene podatke zbira in omogoča nadaljnjo analizo. Na podlagi zbranih podatkov naj diplomsko delo predstavi primere uporabe takšnih podatkov in predvsem identifikacijo poslovnih priložnosti, ki jih takšno zbiranje podatkov prinaša. Pomemben poudarek naj bo tudi na pravnih in etičnih vidikih takšnega sledenja uporabnikom.

Mentor:
doc. dr. Dejan Lavbič

Dekan:
prof. dr. Nikolaj Zimic

Fakulteta za računalništvo in informatiko
2
UNIVERZA V LJUBLJANI

IZJAVA O AVTORSTVU DIPLOMSKEGA DELA

Spodaj podpisani Sašo Sotlar, z vpisno številko **63040369**, sem avtor diplomskega dela z naslovom:

Sledenje uporabnikom mobilnih naprav.

S svojim podpisom zagotavljam, da:

- sem diplomsko delo izdelal samostojno pod mentorstvom dr. Dejana Lavbiča,
- so elektronska oblika diplomskega dela, naslov (slov., angl.), povzetek (slov., angl.) ter ključne besede (slov., angl.) identični s tiskano obliko diplomskega dela,
- soglašam z javno objavo elektronske oblike diplomskega dela v zbirki "Dela FRI".

V Ljubljani, dne 10. oktobra 2013 Podpis avtorja:

Zahvala

Zahvaljujem se staršem, vsem najbližjim in prijateljem, ki so verjeli vame v času mojega šolanja na Fakulteti za računalništvo in informatiko.

Posebej gre zahvala mojemu dekletu Alji, ki mi je vedno stala ob strani in me spodbujala.

Iskrena hvala mentorju, dr. Dejanu Lavbiču, za strokovno pomoč, nasvete in vodenje pri izdelavi, ter prof. Nevenki Matelič-Nunčič za pomoč pri lektoriranju diplomskega dela.

Kazalo

Kazalo slik:

Povzetek

Cilj diplomskega dela je predstaviti način sledenja uporabnikom brezžičnih omrežij in prikazati primere analiz zajetih podatkov. S pomočjo usmerjevalnika, dodatne programske opreme in osnovnih delcev vzpostavljanja povezave v brezžičnem omrežju, okvirjev poskus zahteve, smo izdelali rešitev, ki nam omogoča zajem in zajete podatke obdela z vrsto analiz. Izvedli smo eksperiment zajema podatkov in prikazali primere analiz ter predlagali primere uporabe sistema v realnem okolju. Predstavili smo tudi koncept določanja uporabnikove lokacije na podlagi trilateracije. Nekatere pridobljene podatke, kot je MAC naslov, ki je unikaten, se v nekaterih primerih lahko upošteva tudi kot osebni podatek, saj je s pravim pristopom omogočeno tudi določanje identitete posameznika. V diplomskem delu smo se tako posvetili tudi pravnemu vidiku takšnega početja in opomnili na problematiko varovanja osebnih podatkov, ki se v današnjih časih zbirajo skoraj na vsakem koraku.

Ključne besede:

sledenje, brezžično omrežje, usmerjevalnik, poskus zahteve, MAC naslov, trilateracija, določanje identitete, varovanje osebnih podatkov

Abstract

The objective of the thesis is to present a way to track wireless networks users and to show analysis examples of captured data. Using a router, additional software and with aid of basic components of communication when connecting to a wireless network, probe requests, we created a solution that allowed us to capture wireless data and helped us create some visual analysis of that data. As an experiment, we performed collection of test data, created analysis examples of caputred data and we proposed usages of our solution in real world. We also presented the concept of determining user's location with use of trilateration. Some of the captured data, like MAC address, could be classified as personal data as it is possible, with right approach, to determine identity of an individual. Aim of thesis is also to shed some light on legal aspects of such data collection and to remind reader on the issue of personal data protection, which are in these times, collected on almost every step we take.

Key words:

tracking, wireless network, router, probe request, MAC address, trilateration, determing identity, personal data protection

Seznam uporabljenih kratic in simbolov

LAN	Local Area Network; lokalno omrežje
IEEE	Insitute of Electrical and Electronics Engineers; Inštitut inženirjev elektrotehnike in elektronike
MHz	megahertz; enota frekvence
GHz	gigahertz; enota frekvence
Kbps	kilobits per second; hitrost prenosa v kilobitih na sekundo
Mbps	megabits per second: hitrost prenosa v megabitih na sekundo
OSI	Open Systems Interconnection; model zgradbe protokolov
IP naslov	številka, ki natančno določa napravo v omrežju
QoS	quality of service; kakovost zagotavljanja storitve
VPN	virtual private network; virtualno osebno omrežje
RAM	random access memory; delovni pomnilnik
SSI	signal strength indicator; moč sprejetega signala
SSID	service set identifier; ime brezžičnega omrežja
MAC naslov	media access control address; unikaten naslov omrežne naprave
dBm	decibel milliwats; enota moči sprejetega signala

Poglavje 1

Uvod in motivacija

Brezžična lokalna omrežja so povsod okoli nas – v lokalih, nakupovalnih centrih, pri frizerju, na smučiščih. Čedalje več je tudi raznovrstnih naprav in uporabnikov, ki s svetom in drugimi uporabniki komunicirajo ravno preko teh omrežij. Ko se sprehajamo po večjem mestu, nas ob vsakem koraku obletava na stotine, tisoče podatkov, delcev informacij, ki so namenjeni nekomu ali neki napravi. Marsikdo se sprašuje, komu so namenjeni ali kaj piše v posameznem podatku. Če bi imeli ti podatki fizično pojavno obliko, bi jih lahko z roko ujeli in prebrali. Temu pač ni tako, ni pa nujno, da je to tudi nemogoče. Brezžična omrežja imajo svoje prednosti [1]:

- priročnost (ni potrebno, da smo statični oziroma samo na enem mestu),
- enostavna postavitev (kupimo usmerjevalnik in ga priključimo),
- razširljivost (enostavno dodajanje novih uporabnikov).

Imajo pa tudi slabosti:

- varnost (podatke lahko sprejema vsakdo v območju),
- zanesljivost (motnje signalov).

Ena izmed ključnih slabosti brezžičnih omrežij je varnost. Ravno dejstvo, da lahko do brezžičnega omrežja dostopamo od koderkoli (znotraj dosega dostopne točke oziroma usmerjevalnika), je tisto, ki varnost brezžičnih omrežij postavlja pod vprašaj. Če imamo dostopno točko postavljeno nekje v stanovanju, se brezžično omrežje velikokrat razteza tudi izven stanovanja. V nasprotju s klasičnim, žičnim omrežjem, je torej na voljo vsem, ki so v dosegu delovanja. Tako lahko vsakdo z ustreznim znanjem prestreza in prebira podatke na našem omrežju.

Obstaja tudi obratna možnost – da z našo dostopno točko prebiramo podatke tistih, ki sploh (še) niso na našem brezžičnem omrežju, temveč le v območju delovanja. Vsaka naprava, ki omogoča brezžično povezovanje in ima to možnost vklopljeno, ob iskanju brezžičnih omrežij oddaja nekatere svoje podatke, ki jih lahko prestrežemo in preberemo.

Tako prebrani podatki nam sami ne zmorejo povedati veliko. Da bi lahko iz njih pridobili kakšno koristno informacijo, jih je potrebno prebirati dlje časa, jih shraniti in nato analizirati. Analiza je odvisna predvsem od tega, kdo smo in katere informacije nas zanimajo. Če bi bili trgovci, bi nas zelo zanimalo, kdaj, ob katerih urah v dnevu, tednu, je bil obisk v trgovini največji. Na podlagi pridobljene informacije bi se lahko odločili ukrepati, če bi bilo to

potrebno.

Namen diplomskega dela je tako predstaviti enega izmed načinov, kako lahko te podatke pridobimo in preberemo.

V drugem poglavju najprej na kratko opišemo zgodovino in razvoj brezžičnih omrežij, nato še njihovo delovanje. Opisano je tudi, kako brezžični vmesniki uporabnikov (naprave) brezžična omrežja iščejo in se z njimi povezujejo.

Nato sledi v tretjem poglavju opis tehnologij, s katerimi smo si pomagali tako pri pripravi našega usmerjevalnika za zajem podatkov, kot tudi pri pretvorbi teh podatkov v obliko, ki je primerna za analizo.

V nadaljevanju se posvetimo povezavi teh tehnologij med seboj in njihovi uporabi ter opišemo postopke, s katerimi smo implementirali zajem, pretvorbo in obdelavo podatkov, o katerih govorimo.

V poglavju pet preverimo, kakšni so bili rezultati našega eksperimenta – usmerjevalnik, ki smo ga s postopki, opisanimi v prejšnjih poglavjih, pripravili za zajem, smo postavili na izbrano lokacijo in ga tam pustili, da je zbral zadostno količino podatkov za analiziranje. Ob primerih si ogledamo nekatere izmed možnih analiz, ki jih je s takšnim zajemom mogoče doseči.

Sledi predstavitev možnosti uporabe takšnega sistema v realnih okoljih in pravnih vidikov takšnega početja, nato pa v zadnjem poglavju opišemo še smernice za nadaljnji razvoj sistema.

Poglavje 2

Brezžično omrežje

Brezžično lokalno omrežje (Wireless LAN, v nadaljevanju brezžično omrežje) je povezava dveh ali več računalnikov brez uporabe kablov. Omrežni podatki, ki se navadno prenašajo po kablih, se tako prenašajo s pomočjo radijskih valov.

2.1 Zgodovina brezžičnih omrežij

Prvo brezžično omrežje je bilo vzpostavljeno leta 1971 na Havajski univerzi - imenovalo se je ALOHAnet in je vključevalo sedem računalnikov na štirih otokih, ki so z glavnim računalnikom komunicirali brez uporabe telefonskih žic. Standard, na katerem so zasnovana današnja brezžična omrežja, 802.11, pa ima svoje začetke v letu 1985, ko je Ameriška zvezna agencija za komunikacije sprostila določeno območje radijskega spektra za prosto uporabo.

Prva generacija brezžičnih omrežij je uporabljala območje radijskega spektra med 902 in 928 MHz in je delovalo pri hitrosti 500 Kbps

Proti koncu devetdesetih let prejšnjega stoletja je organizacija IEEE zasnovala skupino za določitev standardov za standard 802.11, katero je vodil Vic Hayes, danes znan kot »oče« brezžičnih omrežij.

Druga generacija je že delovala na frekvenci 2,4 GHz s hitrostjo do 2 Mbps. Tretja generacija temelji na enaki frekvenci kot druga in je v uporabi še danes. [2]

Trenutno najbolj razširjena implementacija brezžičnih omrežij temelji na standardu 802.11n, ki omogoča teoretične hitrosti prenosa do 150 Mbps. V izdelavi je tudi nov standard 802.11ac, ki naj bi podpiral hitrosti do 450 Mbps. [3]

2.2 Delovanje brezžičnih omrežij

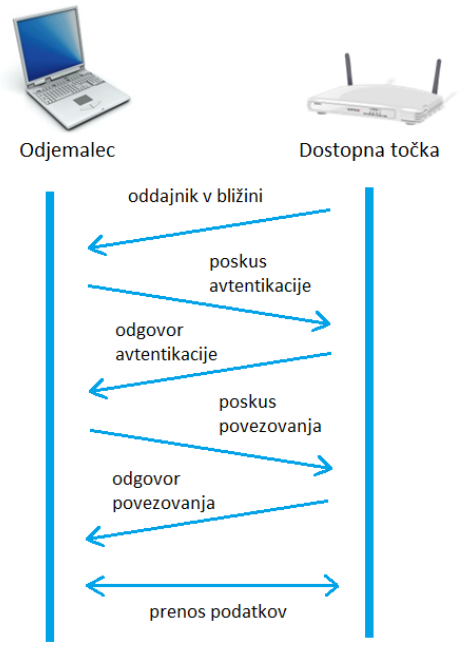
Brezžično omrežje navadno sestavljajo:

- internetna povezava,
- usmerjevalnik / dostopna točka,
- odjemalci / uporabniki (clients).

Slika 1: Elementi brezžičnega omrežja

Povezovanje z brezžičnim omrežjem poteka v več korakih in še predno lahko v našem najljubšem brskalniku poiščemo recept za juho, se med dostopno točko in našo napravo izmenja kar nekaj podatkov. Tem podatkom pravimo okvirji (frames). Povezovanje z brezžičnim omrežjem lahko opišemo kot zaporedje aktivnosti, ki si sledijo v naslednjem vrstnem redu:

1. dostopna točka oddaja signale, s katerimi sporoča napravam, da je v bližini,
2. ko odjemalec dobi informacijo o bližini dostopne točke, se poskusi avtenticirati,
3. če dostopna točka napravo uspešno avtenticira, se lahko začne povezovanje,
4. ko je povezovanje uspešno, odjemalec lahko uporablja brezžično omrežje.



Slika 2: Povezovanje naprave z brezžičnim omrežjem

2.3 Okvirji v brezžičnih omrežjih

Okvir lahko definiramo kot podatek, ki ga ustvari brezžični vmesnik. Okvirji so osnovne enote podatkov na povezovalni (drugi) plasti OSI modela [4], vsebujejo informacije o uporabljenih protokolih, formatih sporočil, mehanizmih dostopa do prenosnega medija itd.

V brezžičnih omrežjih poznamo več vrst okvirjev:

- podatkovni (data frames),
- kontrolni (control frames),
- obvladovalni (management frames).

Podatkovni okvirji prenašajo podatke iz višjih plasti omrežnega modela. Kontrolni okvirji skrbijo za pravilen prenos podatkovnih okvirjev in priskrbijo dostop do prenosnih medijev. Obvladovalni okvirji služijo pri iskanju, identifikaciji in vzpostavljanju povezave med dostopno točko in odjemalcem. Nekateri izmed obvladovalnih okvirjev so:

- okvir oddajnika (beacon),
- poskus zahteve (probe request),
- poskus odgovora (probe response),
- zahteva za avtentikacijo (authentication request),
- avtentikacijski odgovor (authentication response).

Struktura glave posameznih obvladovalnih okvirjev je vedno enaka, podatkovna polja pa so različna glede na podtip obvladovalnega okvirja. [5]

2.4 Načini preiskovanja okolice za brezžična omrežja

Da se lahko odjemalec poveže z brezžičnim omrežjem, mora najprej pregledati, ali je sploh kakšno v bližini.

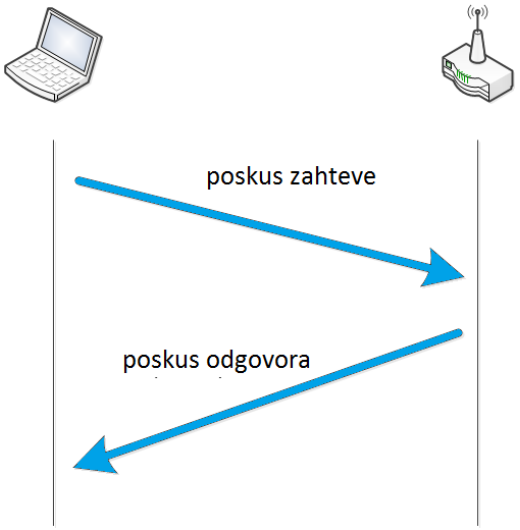
2.4.1 Pasivno iskanje

Pri pasivnem iskanju odjemalec le čaka na okvirje oddajnika (beacon), ki jih v (rednih) časovnih presledkih oddaja dostopna točka. V okvirju oddajnika so zapisane informacije o imenu in zmogljivosti brezžičnega omrežja ter tudi o parametrih, potrebnih za vzpostavitev povezave.

Slika 3: Pasivno iskanje

2.4.2 Aktivno iskanje

Pri aktivnem iskanju odjemalec sam želi izvedeti, ali je v bližini kakšna dostopna točka, ki oddaja brezžično omrežje. Tako v intervalih oddaja okvirje poskus zahteve, na katere tiste dostopne točke, ki so v dosegu, odgovorijo z okvirjem poskus odgovora.



Slika 4: Aktivno iskanje

Aktivno iskanje je v uporabi predvsem zato, ker lahko na tak način hitreje pridobimo podatke o brezžičnih omrežjih okoli nas. Pri pasivnem iskanju smo odvisni od časovnih intervalov, v katerih dostopna točka okvirje oddajnik v bližini oddaja.

2.5 Okvir poskus zahteve

Okvirje poskus zahteve oddaja odjemalec. V podatkovnem polju okvirja sta zapisani dve informaciji:

- seznam podprtih hitrosti prenosa,
- SSID polje.

Da se lahko odjemalec uspešno poveže z dostopno točko, mora ta podpirati vse hitrosti prenosa, katere brezžično omrežje zahteva.

V SSID polju je lahko zapisano ime specifičnega brezžičnega omrežja, na katerega je okvir naslovljen, lahko pa je naslovljen na katerokoli primerno omrežje v okolici (broadcast). Vrednosti SSID polja so lahko imena tistih brezžičnih omrežij, s katerimi se je odjemalec v preteklosti že povezoval. [6]

Poglavje 3

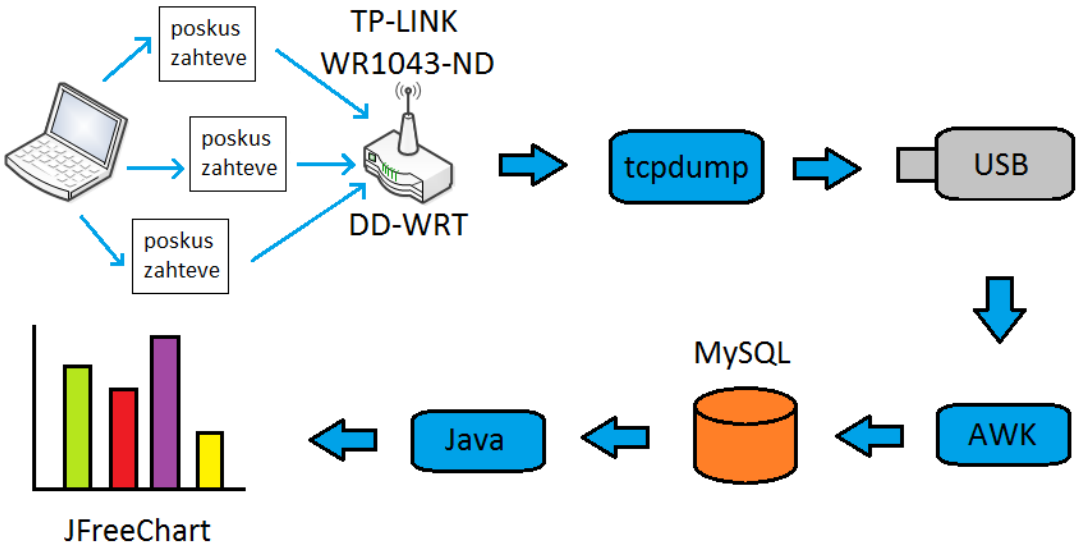
Uporabljene tehnologije testnega okolja

Za zajem podatkov - okvirjev poskus zahteve - ki jih potrebujemo v okviru diplomskega dela, moramo najprej vzpostaviti okolje, v katerem lahko takšne podatke dobimo.

Kakor je bilo že prej predstavljeno, se okvirji poskus zahteve izmenjujejo med dostopno točko in odjemalcem (mobilni telefon, tablica, prenosnik itd.). Zajem okvirjev je tako potekal na izbranem usmerjevalniku TP-LINK WR1043-ND. Ker privzeto nameščena strojna programska oprema na usmerjevalniku (firmware) takšnega zajema ne omogoča, smo nanj namestili alternativno (3rdParty) programsko opremo DD-WRT, ki bazira na Linuxu. Tako je, z določenimi modifikacijami, možno tudi nameščanje dodatnih programskih paketov (tcpdump), katere smo potrebovali. Tako smo lahko shranjevali celotne okvirje poskus zahteve na dodatni zunanji USB disk.

Ko smo pridobili sezname okvirjev, je bilo najprej potrebno iz njih izluščiti tiste informacije, katere smo kasneje potrebovali za analizo. Pri tem nam je bil v pomoč programski jezik AWK. Za izvrševanje poizvedb smo tako pridobljene informacije zapisali v podatkovno bazo MySQL, ki smo jo ustvarili s pomočjo paketa XAMPP, rezultate v programski rešitvi, napisani v programskem jeziku Java, pa prikazali ali v obliki teksta ali pa v obliki grafov, katere smo generirali s pomočjo knjižnice JFreeChart.

Na sliki 5 so prikazane odvisnosti uporabljenih tehnologij v testnem okolju.



Slika 5: Odvisnosti uporabljenih tehnologij v testnem okolju

3.1 Usmerjevalnik TP-LINK WR1043-ND

Usmerjevalnik podjetja TP-LINK [7] smo izbrali zaradi njegovih tehničnih zmogljivosti:

- procesor Atheros AR9132@400MHz,
- 8 MB flash pomnilnika,
- 32 MB RAM pomnilnika,
- USB podpora.

Na flash pomnilniku je prostor za strojno programsko opremo (firmware). Ta ob izpadu električne energije ne izgubi podatkov, katere trenutno hrani. Ker je pomnilnik prepisljiv, lahko nanj nameščamo posodobitve že nameščene strojne programske opreme, lahko pa namestimo tudi katero izmed alternativnih strojnih programskih oprem.

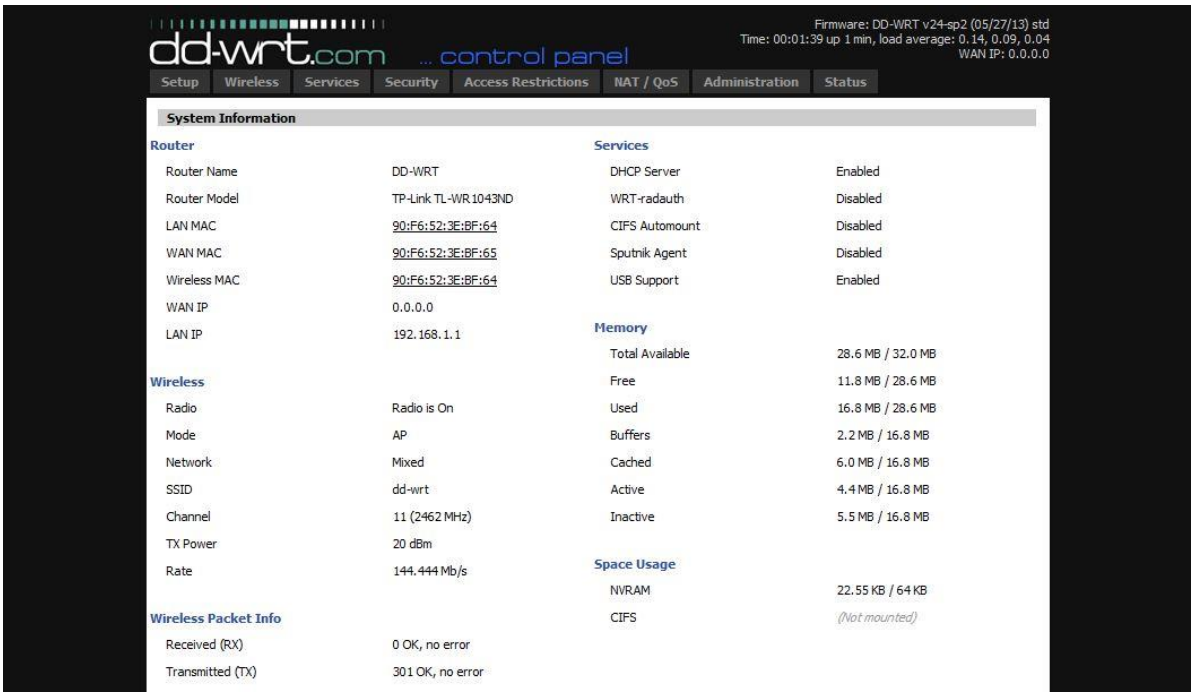
Na drugi strani RAM pomnilnik ob odklopu oz. izpadu električne energije izgubi vse podatke, ki so trenutno zapisani v njem. Je zelo hiter pomnilnik, ki je v usmerjevalniku na voljo strojni programski opremi za vse njene potrebe po branju in pisanju, za njene sistemske tabele in medpomnilnike. V njem se shranjujejo tudi usmerjevalne tabele, ARP (address resolution protocol; protokol za prepoznavanje naslovov) tabele, izvaja se medpomnjenje prispelih in odhajajočih paketov.

Ker je prostor, ki nam je na voljo za zapisovanje naših podatkov, premajhen za naše potrebe, ga je potrebno razširiti. Za to lahko uporabimo USB vhod, preko katerega lahko nanj priključimo dodatni zunanji disk.

3.2 DD-WRT

DD-WRT [8] je alternativna odprtokodna strojna programska oprema za brezžične usmerjevalnike, ki bazira na operacijskem sistemu Linux. Izhaja iz projekta OpenWRT, ki je uporabljen še v drugih alternativnih distribucijah strojne programske opreme, kot na primer:

- FreeWRT,
- Gargoyle,
- Bluebox (samo za WRT54GL usmerjevalnike).



Slika 6: Začetna maska DD-WRT

DD-WRT doda usmerjevalniku dodatne funkcionalnosti, katerih načeloma v večini strojne programske opreme, nameščene na usmerjevalniku ob nakupu, ne moremo najti. Obstaja tudi več različnih verzij DD-WRT programske opreme. Delimo jih po velikosti, kakršno zavzamejo na flash pomnilniku, saj imajo različni modeli usmerjevalnikov različne velikosti flash pomnilnika, na katerega strojno programsko opremo lahko namestimo. Verzije, ki so po velikosti manjše, vsebujejo tudi manj funkcij kot tiste večje.

Nekatere izmed mnogih prednosti oz. funkcij, ki jih prinaša DD-WRT:

- napredni QoS,
- vzpostavljanje VPN povezav,
- prilagajanje moči brezžične antene.

Za DD-WRT smo se odločili predvsem zaradi velikega nabora funkcij, zanesljivega delovanja in stabilnosti, dobre podprtosti glede na izbran usmerjevalnik ter relativno enostavnega postopka za namestitev dodatne programske opreme.

3.3 SSH/PuTTY

Da bi usmerjevalnik lahko ustrezno konfigurirali za namestitev dodatne programske opreme in shranjevanje podatkov, moramo najprej pridobiti administrativni dostop do sistemskih

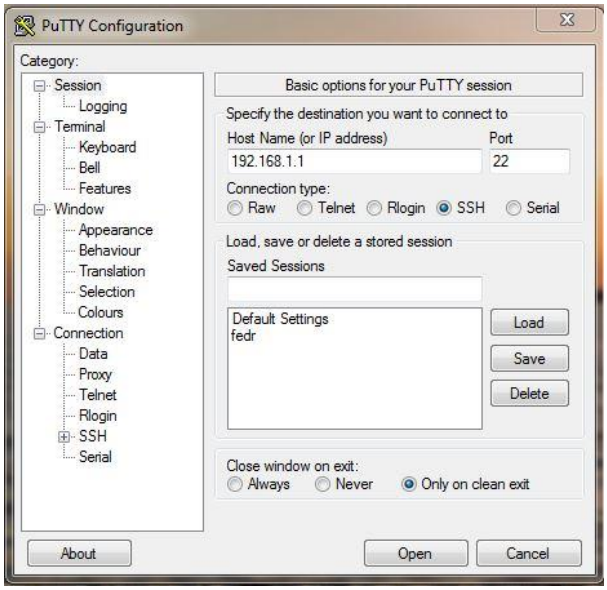
datotek in terminala (dostop do okrnjene Linux lupine v usmerjevalniku).

To storimo preko SSH protokola – **S**ecure **sh**ell. [9] Gre za omrežni protokol, s pomočjo katerega se lahko na napravo (računalnik, usmerjevalnik) varno prijavimo na daljavo. Povezava, ki se vzpostavi, je kriptirana.

Uporabi se lahko princip avtomatskega generiranja javnih in zasebnih ključev za kriptiranje povezave, uporabnik pa se nato prijavi s pomočjo gesla. Ob prijavi se še dodatno preveri, ali ima avtenticirani uporabnik zadostne pravice za dostop do lupine, in se mu na podlagi tega dostop odobri ali zavrne.

Drugi način je, da pare javnih in zasebnih ključev za dostop generiramo sami. V tem primeru nam gesla ni potrebno vpisovati.

Za uporabo SSH protokola potrebujemo odjemalca (client). V operacijskih sistemih Linux je ta že vgrajen v okolje (ukaz »ssh« v terminalu). Za Windows okolje za ta namen obstajajo posebni programski odjemalci – eden takšnih je Putty. (http://www.putty.org)



Slika 7: Zaslonska maska SSH odjemalca PuTTY

3.4 GParted

GParted [10] je orodje za urejanje particij na disku – z njim lahko particije ustvarimo, spreminjamo (krčimo, razširjamo), kloniramo ali kopiramo itd. Podpira veliko datotečnih sistemov, med njimi:

- ext2, ext3, ext4,
- FAT16, FAT32,

- HFS, HFS+,
- NTFS.

Na spletni strani projekta pridobimo .iso datoteko (posnetek diska), katero lahko zapišemo na zgoščenko (CD), ob ponovnem zagonu računalnika izberemo zagon sistema iz CD-ROM pogona in pokazal se nam bo seznam z različnimi verzijami programa Gparted. Ko izbrano verzijo zaženemo, bo ta poiskal vse diske, ki so trenutno priključeni na sistem. Izberemo tistega, na katerem želimo kaj spremeniti in nad njim operiramo.

Za naše potrebe smo GParted uporabili tako, da smo na USB disku ustvarili particijo z datotečnim sistemom ext-3.

3.5 tcpdump

Tcpdump [11] je program za analizo mrežnih paketov in omrežij. Z njim lahko prestrežemo in preberemo različne pakete, ki so poslani preko omrežja, o katerem delamo poizvedbo. Uporablja v C/C++ spisano knjižnico libpcap. Deluje v večini operacijskih sistemov, ki bazirajo na Unix-u (OS X, Solaris, BSD, Linux), za operacijski sistem Windows pa obstaja posebej napisana verzija, imenovana WinDump, ki uporablja knjižnico WinPcap (prav tako posebej napisana verzija knjižnice libpcap za Windows okolje).

Tipična uporaba programa tcpdump obsega analizo omrežja, njegovo učinkovitost ter vpogled v obnašanje aplikacij, ki na tem omrežju komunicirajo. Z njegovo pomočjo lahko tudi ugotavljamo, ali usmerjanje prometa v omrežju deluje tako, kot mora, lahko pa tudi poskušamo odkriti razlog, zakaj ni tako.

S pravilno pripravljeno omrežno arhitekturo ga lahko uporabimo tudi v namene prestrezanja prometa in komunikacije med drugimi uporabniki oz. napravami v omrežju.

Sam program ponuja številne opcije za zajem in filtriranje paketov glede na njihov tip, formatiranje, način izpisovanja itd.

```
22:10:09.237831 27276072us tsft 1.0 Mb/s 2437 MHz 11b -76dB signal antenna 1 BSSID:Broadcast DA:Broadcast
SA:b4:b6:76:5f:e3:58 (oui Unknown) Probe Request () [1.0 2.0 5.5 11.0 6.0 9.0 12.0 18.0 Mbit]
22:10:09.238531 27276929us tsft 1.0 Mb/s 2437 MHz 11b -77dB signal antenna 1 BSSID:Broadcast DA:Broadcast
SA:b4:b6:76:5f:e3:58 (oui Unknown) Probe Request (Sotlar) [1.0 2.0 5.5 11.0 6.0 9.0 12.0 18.0 Mbit]
```

Slika 8: Primer izpisa zajema paketov s tcpdump programom

3.6 iwconfig

Iwconfig [12] je program, s katerim lahko izpišemo brezžične vmesnike, pregledujemo in nastavljamo njihove parametre, lahko pa se ga uporabi tudi za izpis statistike na izbranem brezžičnem vmesniku. Brezžični vmesniki podpirajo šest različnih načinov delovanja [13]:

- glavni (master, vmesnik deluje kot dostopna točka),
- upravljalski (managed, vmesnik se poveže v omrežje, sestavljeno iz več dostopnih točk, gostovanje),
- vsak z vsakim (peer-to-peer, omrežje brez dostopne točke),
- ponavljalni (repeater, vmesnik posreduje podatke med dvema drugima),
- nadzorni (monitoring, vmesnik ne oddaja ničesar, samo pasivno pregleduje promet).

Omrežni vmesnik privzeto deluje tako, da sprejme le tiste pakete, ki so direktno naslovljeni nanj ali pa so poslani po vsem omrežju vsem postajam (broadcast). Druge pakete zavrže. V nadzornem načinu omrežni vmesnik teh paketov ne zavrže, temveč jih sprejme, mi pa jih lahko zajamemo in tako preberemo. Brezžični vmesnik v nadzornem načinu ne oddaja brezžičnega omrežja (SSID).

iwconfig tako potrebujemo za nastavitev načina delovanja brezžičnega vmesnika na usmerjevalniku iz glavnega v nadzorni način.

3.7 AWK (regex)

AWK [14] je programski jezik, ustvarjen z namenom procesiranja in obdelave teksta. Je standardna funkcija večine operacijskih sistemov, ki bazirajo na UNIX-u, velikokrat pa si z njim pomagamo takrat, ko želimo iz nekega teksta pridobiti posamezne informacije oz. dele tega teksta.

Je programski jezik, ki ne potrebuje prevajanja (compiling) kode in tako neposredno izvaja izvorno kodo. Razvit je bil v Bell-ovih laboratorijih, ime pa nosi po svojih avtorjih: Alfred **A**ho, Peter **W**einberger, Brian **K**ernighan. Kot vhod mu lahko podamo tekstovno datoteko, lahko pa tudi preusmerjen izhod drugega programa (pipeline).

Z AWK programskim jezikom lahko počnemo marsikaj: od izpisovanja teksta po poljih, iskanja po vzorcih, operiramo nad izbranim tekstom, lahko vpeljemo vejitvene stavke (if statements), zanke, spremenljivke, prilagajamo izpis...

Standardna oblika AWK programa je:

```
BEGIN               {<incializacija>}
/iskani vzorec 1/ {<akcije>}
/iskani vzorec 2/ {<akcije>}
END                 {<zakljucne akcije>}
```

Iskani vzorci so lahko regularni izrazi. Regularni izraz je mehanizem za opisovanje sestave besedila, ki izvaja iskanje, zamenjavo in iskanje teksta v datotekah. Vsebuje navadne znake

oz. črke, ki pa imajo določen pomen. Na primer znak . (pika) predstavlja katerikoli znak v nizu. Znaki *, ?,+ se uporabljajo kot operatorji ponavljanja, itd. Tako so lahko regularni izrazi v kombinaciji z AWK programskim jezikom zelo močno in uporabno orodje pri obdelavi teksta.

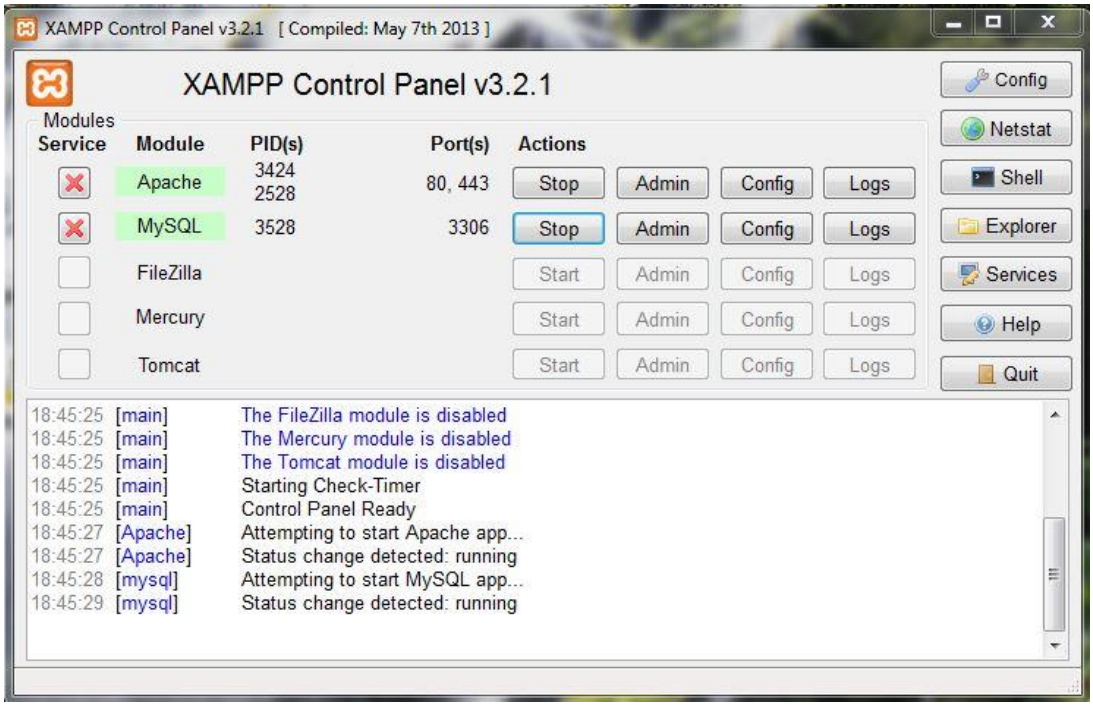
3.8 XAMPP

XAMPP [15] je neplačljiv, odprtokoden paket programske opreme, ki se uporablja za zagotavljanje popolnoma funkcionalne rešitve spletnega strežnika. Paket klasificiramo kot LAMP distribucijo – akronim LAMP sestavljajo:

- **L**inux (operacijski sistem),
- **A**pache (spletni strežnik),
- **M**ySQL (strežnik podatkovne baze),
- **P**HP, **P**erl, **P**ython (skriptni jezik).

Po namestitvi paketa in zagonu spletnega strežnika Apache lahko odpremo spletni brskalnik na naslovu http://localhost in odprla se nam bo začetna stran. Tu bomo med drugim našli tudi myPhpAdmin, spletno aplikacijo za MySQL strežnik. V tej aplikaciji lahko enostavno ustvarimo novo podatkovno bazo, tabele, pregledujemo podatke v tabelah itd.

V aplikaciji myPhpAdmin smo tudi ustvarjali testne poizvedbe nad testnimi podatki, katere smo potem lahko implementirali v programski rešitvi.



Slika 9: Zaslonska maska XAMPP paketa

3.9 JFreeChart

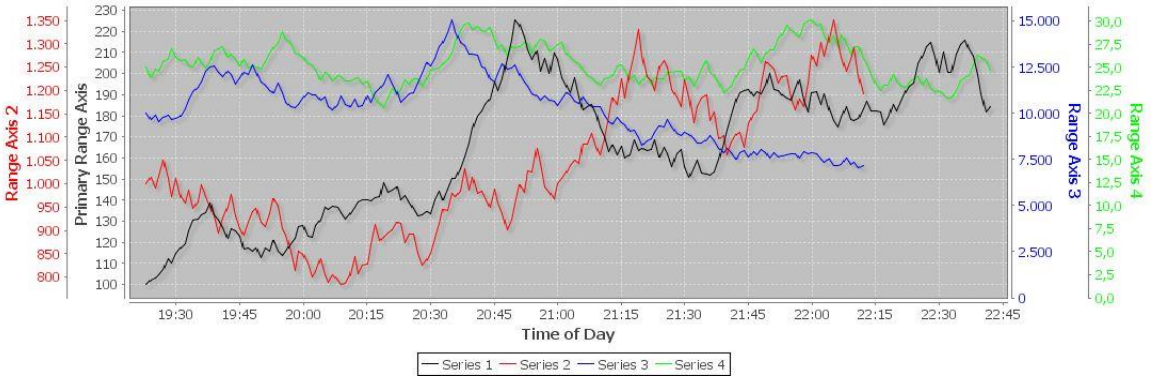
JFreeChart [16] je odprtokodno ogrodje za programski jezik Java, ki omogoča enostavno generiranje tako interaktivnih kot tudi neinteraktivnih diagramov. Projekt je začel David Gilbert februarja leta 2000. Od takrat dalje je JFreeChart postala ena izmed najbolj uporabljenih tovrstnih knjižnic za Javo. Projekt se še vedno razvija, tako s pomočjo ustanovitelja kot tudi z delom prostovoljcev.

JFreeChart vsebuje zelo podrobno spisano dokumentacijo, njegov dizajn je zelo fleksibilen, kar omogoča razširljivost in prilagodljivost, podpira več načinov izrisovanja diagramov (na primer izvoz v slikovne datoteke JPEG, PNG, vektorsko grafiko SVG, EPS, tudi PDF) in je brezplačen, ker je izdan pod licenco LGPL (GNU Lesser General Public Licence). Lahko se ga uporabi tudi v plačljivih aplikacijah.

Podpira 14 tipov diagramov, med drugim tudi:

- (več)stolpične diagrame,
- tortne diagrame,
- Ganntove diagrame,
- ploščinske diagrame,
- časovne diagrame.

V programski rešitvi smo uporabljali predvsem stolpične diagrame.



Slika 10: Primer naprednega grafa, generiranega s knjižnico JfreeChart

Poglavje 4

Implementacija programske rešitve za analizo Probe Request okvirjev

Programsko rešitev smo implementirali v integriranem razvojnem okolju Netbeans in programskem jeziku Java. Izbirali smo med razvojem v Microsoftovem okolju Visual Studio in s programskim jezikom C#, vendar smo se zaradi enostavnosti, odprtosti in predvsem zmožnosti poganjanja programa na različnih platformah odločili za Javo. Navsezadnje je kar nekaj postopkov potrebno izvesti v Linuxu (sicer obstajajo tudi alternative za okolje Windows, vendar jih je v večini potrebno dodatno namestiti, npr. WinDump (alternativa programu tcpdump), WinPcap (alternativa knjižnici libpcap), Gawk for Windows (AWK programski jezik za Windows, itd.), tako da je smiselno imeti spisan program, ki bo tekel tudi v Linux okolju.

Izvorna koda programske rešitve je javno dostopna v spletnem repozitoriju GitHub na naslovu: https://github.com/sash69/wlanAnalysis

4.1 Namestitev DD-WRT na usmerjevalnik

Za nameščanje alternativne strojne programske opreme na usmerjevalnik je dobro imeti zaledje oziroma ustrezno predznanje. Večina proizvajalcev zato neukim uporabnikom ter tistim, ki ne vedo, čemu alternativno strojno programsko opremo sploh potrebujejo, nameščanje odsvetuje, saj se lahko zaradi nepravilnega postopka oziroma ob nedoslednem upoštevanju navodil kaj hitro zgodi, da postane usmerjevalnik neuporaben (v računalniškem žargonu - bricked). Ob nameščanju alternativ se dejansko zanašamo na znanje in programerske sposobnosti nekoga oziroma neke skupnosti, kar pa ni nujno stoodstotno zanesljivo. Ko dobimo usmerjevalnik iz trgovine, z veliko zanesljivostjo vemo, da ta deluje (razen če je v proizvodnji prišlo do kakšne stvarne napake), z nameščanjem alternativ pa to ni nujno. Na uradni strani projekta obstaja veliko navodil in nasvetov, kako se lotiti nameščanja DD-WRT programske opreme na usmerjevalnik, vsem pa je skupno to, da se je potrebno *dosledno* držati vseh navodil, drugače lahko usmerjevalnik (tudi nepopravljivo) poškodujemo.

Najprej je potrebno s spletne strani projekta (oz. na naslovu ftp://dd-wrt.com/others/eko/BrainSlayer-V24-preSP2) pridobiti namestitveno datoteko z DD-WRT strojno programsko opremo. Zelo pomembno je, da izberemo pravilno verzijo za naš usmerjevalnik. Na izbranem naslovu izberemo najprej leto, nato ustrezno različico (build; ob času nameščanja na usmerjevalnik, je bila najnovejša različica 05-27-2013-r21676). Priporočljivo je tudi najprej pregledati forum na spletni strani projekta, da se seznanimo z morebitnimi težavami posameznih različic, kajti zgodi se, da kakšna izmed njih povzroča

težave določenim usmerjevalnikom – zato ni nujno potrebno, da je zadnja izdana različica tudi najboljša izbira. V seznamu usmerjevalnikov poiščemo našega, znotraj te mape pa sta na voljo dve namestitveni datoteki:

- factory-to-ddwrt.bin,
- tl-wr1043nd-webflash.bin.

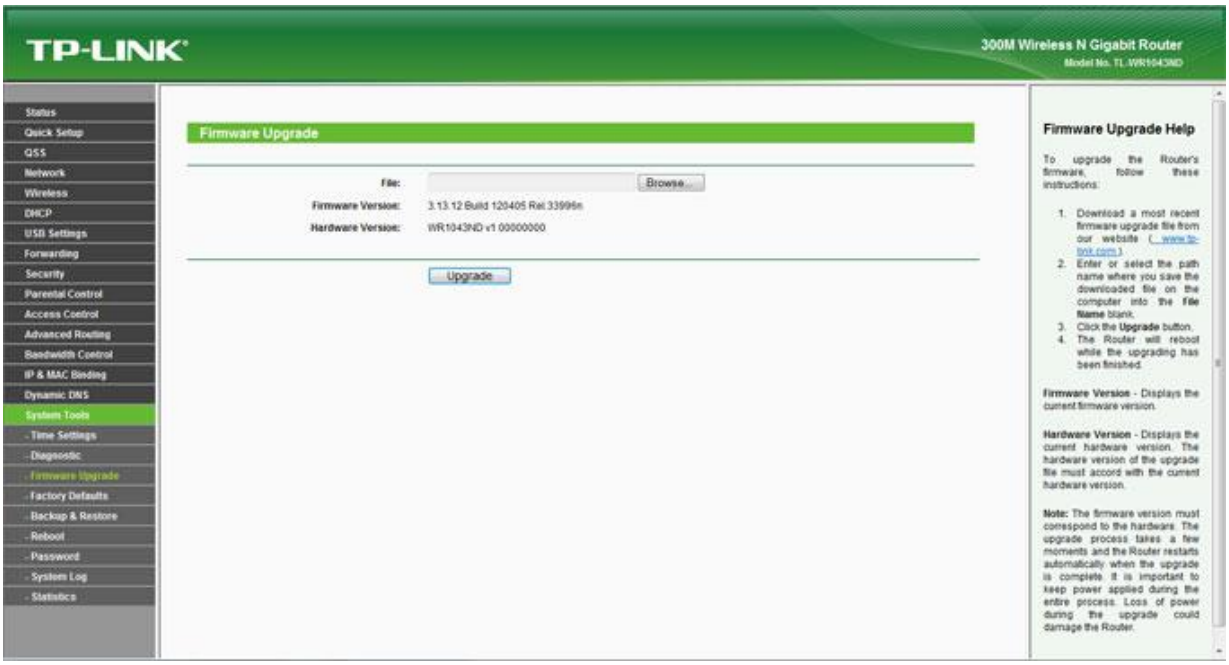
Če imamo na usmerjevalniku nameščeno originalno strojno programsko opremo (factory firmware), potem izberemo namestitveno datoteko factory-to-ddwrt.bin in jo shranimo na poljubno mesto na našem trdem disku. Ta se razlikuje od namestitvene datoteke tl-wr1043nd-webflash.bin po tem, da ima spremenjen način namestitve, ki upošteva tovarniški zagonski nalagalnik (bootloader) in ga tako lahko pravilno spremeni. Druga datoteka je za primer, ko imamo na usmerjevalniku že nameščen DD-WRT (starejša različica), z njo ga lahko posodobimo.

Pred nameščanjem je potrebno usmerjevalnik »totalno ponastaviti« (hard reset, tudi: reset 30/30/30) – to je postopek, ki izbriše vse podatke iz NVRAM pomnilnika (zelo hitri pomnilnik, kjer se shranijo zagonske nastavitve), ter vse nastavitve ponastavi na tovarniške. Izvedemo ga v treh korakih:

1. ko je usmerjevalnik še prižgan, pritisnemo in držimo gumb za ponastavitev 30 sekund,
2. ne da bi spustili gumb za ponastavitev, usmerjevalnik izklopimo iz električnega napajanja in držimo gumb za ponastavitev še naslednjih 30 sekund,
3. medtem ko držimo gumb za ponastavitev, usmerjevalnik priklopimo nazaj na električno omrežje in gumb za ponastavitev držimo še zadnjih 30 sekund.

Usmerjevalnik žično povežemo z našim računalnikom, odpremo spletni brskalnik, v naslovno vrstico vpišemo 192.168.1.1 (oziroma IP naslov usmerjevalnika v našem omrežju). Pojavi se nam maska za vnos uporabniškega imena in gesla (v prejšnjem koraku smo usmerjevalnik totalno ponastavili – s tem sta se ponastavila tudi uporabniško ime in geslo, ki imata sedaj privzete vrednosti, te so zapisane v navodilih usmerjevalnika), nato pa še spletni vmesnik. V meniju izberemo sistemske nastavitve (System tools) ter ukaz nadgradnja strojne programske opreme (Firmware upgrade), izberemo namestitveno datoteko, ki smo jo pridobili v prejšnjem koraku, in počakamo, da se postopek zaključi.

Zelo pomembno je, da ne prekinjamo namestitvenega postopka na noben način. Postopek navadno potrebuje od 2 do 5 minut, da se uspešno zaključi. Po uspešni namestitvi se nam prikaže maska za vnos novega administratorskega uporabniškega imena in gesla.



Slika 11: Spletni vmesnik usmerjevalnika in maska za izbiro nove strojne programske opreme

Ko vnesemo novo uporabniško ime in geslo, na usmerjevalniku še enkrat izvedemo postopek »totalnega ponastavljanja«.

4.2 Povezovanje z usmerjevalnikom preko SSH

Preden z odjemalcem Putty dostopamo do usmerjevalnika preko SSH protokola, je potrebno SSH dostop omogočiti na samem usmerjevalniku, saj je privzeto takšen dostop onemogočen. Prijavimo se v spletni vmesnik usmerjevalnika, v glavnem meniju izberemo zavihek storitve (Services), na maski, ki se pojavi, poiščemo »Secure shell« ter izbiro omogočimo (enable). Nastavitve shranimo in usmerjevalnik ponovno zaženemo.

Slika 11: Omogočanje SSH dostopa do usmerjevalnika

V Windows okolju zaženemo odjemalca Putty, v začetno masko v polje naslov gostitelja (Host name (or IP address)), vnesemo IP naslov routerja (v našem primeru 192.168.1.1), vse ostale nastavitve pa pustimo. S klikom na gumb odpri povezavo (Open) se nam odpre novo terminalsko okno, ki nas povpraša po uporabniškem imenu in geslu: kot uporabniško ime vpišemo »root«, geslo pa uporabimo tisto, katero uporabljamo za prijavo v spletni vmesnik strežnika. Po avtentikaciji se nam izpišejo informacije o lupini.

Slika 12: DD-WRT lupina

4.3 Ureditev dodatnega prostora na usmerjevalniku za namestitev dodatne programske opreme in shranjevanje podatkov

USB disk, ki ga bomo uporabili za namestitev dodatne programske opreme in na njem shranjevali zajete podatke, moramo najprej primerno pripraviti. Pomagamo si s prej omenjenim programom GParted, s katerim na USB disku izbrišemo vse obstoječe particije in ustvarimo novo z datotečnim sistemom ext-3.

Tako pripravljen USB disk priključimo na USB vhod na usmerjevalniku. Na usmerjevalniku je sedaj potrebno omogočiti USB naprave. V glavnem meniju spletnega vmesnika DD-WRT izberemo zavihek storitve (Services), v podmeniju »USB«, ter tam omogočimo:

- podpora USB napravam (Core USB support),
- podpora USB diskovnim napravam (USB storage support),
- samodejna priprava diska za branje (Automatic drive mount),
- lokacija v obstoječem datotečnem sistemu, kjer naj se disk nahaja – iz spustnega menija izberemo »/mnt« (Disk mount point).

Nastavitve shranimo in usmerjevalnik ponovno zaženemo.

Nato je potrebno pripraviti strukturo za namestitev dodatne programske opreme na naš USB

disk. Najprej je potrebno preveriti, če je usmerjevalnik zaznal in pripravil USB disk. To preverimo v dveh korakih:

1. v glavnem meniju spletnega vmesnika izberemo zavihek storitve (Services), v podmeniju pa »USB«. Pod oznako informacije o disku (Disk info) se izpišejo podatki o disku – to pomeni, da je usmerjevalnik USB disk zaznal;
2. zaženemo PuTTY in se prijavimo v usmerjevalnik. Poiščemo direktorij, v katerem je DD-WRT pripravil (mount) naš disk. V terminal vpišemo naslednje ukaze:

```
cd /mnt
ls
```

Izpiše se ime direktorija, v katerem je disk pripravljen: sda_part1

4.4 Namestitev dodatne programske opreme na DD-WRT

Ko imamo pripravljen USB disk in prostor, na katerega lahko prosto zapisujemo, nadaljujemo namestitev programa opkg [17] – programa za upravljanje paketov na integriranih operacijskih sistemih – kakršen je DD-WRT.

V PuTTY terminalskem oknu nadaljujemo z ukazi:

```
cd /sda_part1
#izberemo usb disk
mkdir etc opt root
mkdir /opt/lib
#ustvarimo novo strukturo
chmod 755 etc opt root
chmod 755 /opt/lib
#nastavimo primerne pravice
cp -a /etc/* /mnt/sda_part1/etc
#vse kar je v sistemskem /etc direktoriju skopiramo na disk
mount -o bind /mnt/sda_part1/etc /etc
# /etc naj kaže na diskovni /etc
mount -o bind /mnt/sda_part1/opt /jffs
# /jffs naj kaže na diskovni /opt
```

Ko je struktura na USB disku pripravljena, namestimo opkg program:

```
cd /tmp
wget
http://downloads.openwrt.org/snapshots/trunk/ar71xx/packages/l
```

```
ibc_0.9.33.2-1_ar71xx.ipk
#prenesemo paket libc
wget
http://downloads.openwrt.org/snapshots/trunk/ar71xx/packages/o
pkg_618-5_ar71xx.ipk
#prenesemo paket opkg
ipkg install libc_0.9.33.2-1_ar71xx.ipk opkg_618-5_ar71xx.ipk
#oba paketa namestimo
```

Za opkg program ustvarimo še datoteko z nastavitvami:

```
cat > /etc/opkg.conf << EOF
src/gz snapshots
http://downloads.openwrt.org/snapshots/trunk/ar71xx/packages
#lokacija repozitorija
dest root /opt
#nastavimo »root« (privzeto) destinacijo namestitev na /opt
dest ram /opt/tmp
#nastavimo »ram« destinacijo namestitev na /opt
lists_dir ext /opt/tmp/var/opkg-lists
#direktorij v katerega se shrani seznam paketov iz
repozitorija
EOF
```

Preverimo, če namestitev programa opkg deluje:

```
umount /jffs
#direktorij s knjižnjicami potrebujemo samo za namestitev opkg
mount -o bind /mnt/sda_part1/root /tmp/root
# tmp/root naj kaže na /root direktorij na USB disku
mount -o bind /mnt/sda_part1/opt /opt
# opt naj kaže na /opt direktorij na USB disku
export LD_LIBRARY_PATH='/opt/lib:/opt/usr/lib:/lib:/usr/lib'
#nastavimo pot do knjižnjic
opkg update
#izvedemo posodobitev lokalnega seznama paketov repozitorija
```

Ko pridobimo seznam paketov iz repozitorija, tega lahko prikažemo z ukazom:

```
opkg list
```

Pri izpisovanju in iskanju primernih paketov si lahko pomagamo s preusmerjanjem:

```
opkg list | grep <vzorec>
```

Najprej namestimo še knjižnico, ki jo potrebuje večina programov:

```
cd /tmp
wget
http://downloads.openwrt.org/snapshots/trunk/ar71xx/packages/l
ibc_0.9.33.2-1_ar71xx.ipk
opkg install libc_0.9.33.2-1_ar71xx.ipk
```

Nato namestimo še tcpdump.

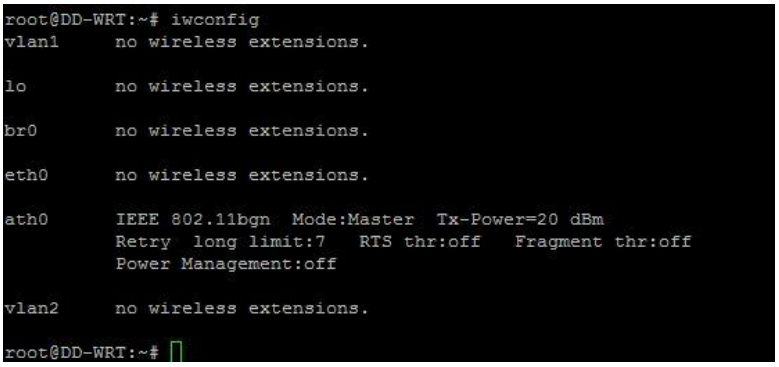
```
opkg install tcpdump
```

4.5 Ureditev bash skripte za nastavitve usmerjevalnika ob zagonu in zajem podatkov

Nato je potrebno nastaviti brezžični vmesnik na usmerjevalniku na nadzorni način. Najprej preverimo, kakšno je ime našega brezžičnega vmesnika:

```
iwconfig
```

S tem ukazom dobimo seznam brezžičnih vmesnikov na našem usmerjevalniku:



Slika 14: Seznam brezžičnih vmesnikov z njihovimi parametri

Naš brezžični vmesnik nosi ime »ath0«. Iz izpisa ukaza iwconfig opazimo tudi parameter način delovanja (Mode), ki je nastavljen na vrednost »Master« (glavni način). Nadzorni način delovanja na brezžičnem vmesniku nastavimo:

```
ifconfig ath0 down              #izključimo brezžični vmesnik
iwconfg ath0 mode Monitor       #spremenimo način delovanja
ifconfig ath0 up                #vključimo brezžični vmesnik
```

Če še enkrat uporabimo ukaz iwconfig, bo parameter način delovanja (Mode) pri brezžičnem vmesniku »ath0« sedaj nastavljen na »Monitor«. Sedaj lahko testiramo, ali brezžični vmesnik sprejema pakete:

```
tcpdump -i ath0 subtype probe-req
```

Če omogočimo brezžično omrežje na neki napravi (npr. telefonu), bomo opazili, da bo vmesnik zajel okvirje poskus zahteve.

Če usmerjevalnik sedaj ponovno zaženemo (ukaz »reboot« v terminalu; izklop in ponoven vklop v električno omrežje), se bo način delovanja brezžičnega vmesnika »ath0« zopet nastavil na glavnega. Tako je potrebno ustvariti skripto, ki ob zagonu usmerjevalnika nastavi brezžični vmesnik in začne zajem.

Najprej kreiramo datoteko optware.enable.

```
cd /mnt/sda_part1
touch optware.enable
```

To datoteko bomo uporabili v skripti kot stikalo – če datoteka obstaja, želimo, da se nastavijo vse poti in knjižnice, potrebne za delovanje naših dodatnih programov, drugače ne naredi ničesar.

V spletnem vmesniku usmerjevalnika v glavnem meniju izberemo zavihek administracija (Administration), v podmeniju pa zavihek ukazi (Commands). V polje za vnos teksta vnesemo naslednjo skripto:

```
#!/bin/sh

 sleep 5
 if [ -f /mnt/sda_part1/optware.enable ]; then
 #če optware.enable obstaja
 mount -o bind /mnt/sda_part1/etc /etc
 mount -o bind /mnt/sda_part1/root /tmp/root
 mount -o bind /mnt/sda_part1/opt /opt
 #pripravimo strukturo na USB disku
 else
 exit
 fi

 if [ -d /opt/usr ]; then
 #če obstaja direktorij usr na USB disku
 export LD_LIBRARY_PATH='/opt/lib:/opt/usr/lib:/lib:/usr/lib'
 export
PATH='/opt/bin:/opt/usr/bin:/opt/sbin:/opt/usr/sbin:/bin:/sbin
```

```
:/usr/sbin:/usr/bin'
 #nastavimo še poti do knjižnjic
 else
 exit
 fi

 date -s 1306280945
 #nastavimo datum v formatu YYMMDDHHMM

 sleep 2
 ifconfig ath0 down
 iwconfig ath0 mode Monitor
 ifconfig ath0 up
 #brezžični vmesnik pripravimo za zajem
 sleep 2
 tcpdump -i ath0 subtype probe-req -G 86400 -w
/tmp/mnt/sda_part1/output%F.cap
```

Zadnjemu ukazu v skripti, tcpdump, smo dodali 2 novi opciji:

- -G 86400
  - opcija se uporablja v navezi z opcijo –w,
  - zajema okvirje za 86400 sekund (24 ur), nato začne pisati v novo datoteko (ali enako, če ime datoteke ostane enako),
- -w /tmp/mnt/sda_part1/output%F.cap
  - –w opcija zapiše zajeti okvir v datoteko na dano lokacijo (/tmp/mnt/sda_part1/) in z imenom output%F.cap,
  - v imenu datoteke %F (output%F.cap) nadomesti trenutni datum.

Tako smo dosegli zajem in shranjevanje okvirjev v datoteke .cap po dnevih. Ko skripto zaženemo, se okvirji shranjujejo v datoteko »outputPRVIDAN.cap«. Po preteklih 24 urah se bodo okvirji začeli shranjevati v datoteko »outputNASLEDNJIDAN.cap«.

Skripto shranimo s klikom na shrani ob zagonu. (Save startup) Skripta se bo vedno pognala ob zagonu usmerjevalnika in prekinila izvajanje v primeru, ko USB disk (s pravilno strukturo) ne bo priključen na usmerjevalnik.

4.6 Pretvorba zajetih podatkov z AWK

Po končanem zajemu je potrebno zbrane okvirje prebrati in iz njih pridobiti le tiste podatke, ki nas zanimajo. Glede na potrebe diplomskega dela smo iz okvirjev izluščili:

- datum (date),
- časovni žig (timestamp),

- MAC naslov izvora (source MAC),
- moč signala (SSI),
- ime brezžičnega omrežja (SSID).

Ker program tcpdump vsebuje zmožnost pisanja zajema v datoteko, ima tudi zmožnost branja takšne datoteke. Iz tcpdump iz datoteke beremo:

```
tpcdump -r <ime_datoteke>
```

Če prebrano izpišemo na zaslon, dobimo podoben izpis (z rdečo pisavo so označeni deli, ki nas zanimajo):

```
2013-10-05 23:28:10.373543 13923646911us tsft 1.0 Mb/s 2412
MHz 11b -76dB signal antenna 1 BSSID:Broadcast DA:Broadcast
SA:9c:4e:36:89:fa:d8 (oui Unknown) Probe Request (HORNIZICE)
[1.0 2.0 5.5 11.0 6.0 9.0 12.0 18.0 Mbit]
```

Datum in točen čas zajema sta vedno na mestih 1 in 2, za ostale podatke pa se lahko pozicija (če štejemo pozicije kot zaporedna mesta med presledki) spreminja. Tu nam pridejo prav AWK in regularni izrazi.

Regularni izrazi za pridobitev:

- moč signala: \-?[0-9]*dB
- MAC naslov izvora: SA:[^:][^:]:[^:][^:]:[^:][^:]:[^:][^:]:[^:][^:]:[^:][^:]
- ime brezžičnega omrežja: t \(.*\) \[

Pri imenu brezžičnega omrežja smo se prepričali, da smo zajeli vse, kar je med znaki:

- t (
-) [

Nato smo tako pridobljenemu nizu odvzeli dva znaka na začetku in dva znaka na koncu in tako dobili celotno ime brezžičnega omrežja (vključno z oklepajema), ki lahko vsebuje vse znake (tudi znak za presledek).

Vse skupaj smo strnili v skripto z imenom »extract.sh«, ki je v istem direktoriju kot vse datoteke z zajemom .cap:

```
#!/bin/bash

directory="parsed" #ciljni direktorij

if [ ! -d "$directory" ]; then
```

```
     mkdir $directory
fi


for i in $(ls);do #za vsako datoteko v direktoriju
    if [[ $i = *.cap ]];then #če je datoteka .cap
          filename=$(basename "$i") #vzamemo ime datoteke
          filename="${filename%.*}" #brez končnice
          tcpdump -e -tttt -r $i |
awk
'{for(i=1;i<=NF;i++) #za vsako polje v vrstici
{
ssi=match($i,/\-?[0-9]*dB/); #preveri ali je ssi
source=match($i,/SA:[^:][^:]:[^:][^:]:[^:][^:]:[^:][^:]:[^:][^
:]:[^:][^:]/); #preveri ali je MAC naslov izvora
if(ssi || source || i==1 || i==2) #če je MAC, ssi, polje 1, 2
     {printf "%s",$i"||"}} #ga izpiši, sledi ||
SSid = match($0,/t \(.*\) \[/); #preveri ali je SSID
if(SSid) #če je, ga izpiši
     {printf "%s", substr($0,SSid+2,RLENGTH-4)}
     #pred izpisom izbriši 2 znaka na začetku in 2 na koncu
{printf "\n"} #postavi se v novo vrstico
}'
> ${directory}/${filename}.parsed
#vse zapiši v datoteko z imenom stare in končnico .parsed
done
```

4.7 Kreiranje MySQL baze in tabele

Bazo smo ustvarili v spletni aplikaciji myPhpAdmin, ki je del programskega paketa XAMPP. Najprej zaženemo XAMPP Control panel, tam pa zaženemo modula Apache in MySQL. Predvsem pri modulu Apache se znajo pojaviti težave, če so omrežna vrata (port) 80 že zasedena (tipične aplikacije, ki lahko zasedajo omrežna vrata 80, so: Skype, Teamviewer itd.), zato je potrebno vrata sprostiti (ugasniti aplikacije in procese, ki ta vrata uporabljajo). Ko sta oba modula zagnana, odpremo spletni brskalnik in v naslovno vrstico vpišemo: http://localhost. Izberemo želeni jezik in odpre se nam osnovna stran XAMPP aplikacije.

myPhpAdmin aplikacijo najdemo v meniju na levi strani. Ustvarimo novo podatkovno bazo z imenom »wlananalysis« in pravilom za razvrščanje znakov »utf8_slovenian_ci«. Podatkovno bazo izberemo in kreiramo tabelo z imenom »requests« s šestimi stolpci:

Ime	Vrsta	Pravilo za razvrščanje znakov	Null	Privzeto	Dodatno
ix	bigint(20)		Ne	Brez	AUTO_INCREMENT
date	date		Da	NULL	
timestamp	timestamp		Da	NULL	
source	varchar(17)	utf8_slovenian_ci	Da	NULL	
ssi	smallint(6)		Da	NULL	
ssid	varchar(100)	utf8_slovenian_ci	Da	NULL	

Ko sta podatkovna baza in tabela ustvarjeni, ju lahko izvozimo in s tem pridobimo že narejene SQL stavke, ki jih kasneje uporabimo v programski rešitvi z algoritmom:

- preverimo, če obstaja baza z imenom »wlananalysis«,
  - če baza ne obstaja, jo kreiramo;
- preverimo, če obstaja tabela z imenom »requests«,
  - če tabela ne obstaja, jo kreiramo.

Podatkovno bazo izvozimo:

1. na prvi strani aplikacije myPhpAdmin izberemo bazo »wlananalysis«,
2. v menijski vrstici izberemo izvozi (Export),
3. izberemo po meri (Custom) način izvoza,
4. med možnostmi izberemo struktura (Structure, privzeto je izbrano »struktura in podatki«),
5. izvoženo lahko shranimo kot datoteko .sql ali samo prikažemo.

SQL stavki za kreiranje tabele »requests«:

```
CREATE TABLE IF NOT EXISTS `requests` (
  `ix` bigint(20) NOT NULL AUTO_INCREMENT,
  `date` date DEFAULT NULL,
  `timestamp` timestamp NULL DEFAULT NULL,
  `source` varchar(17) COLLATE utf8_slovenian_ci DEFAULT NULL,
  `ssi` smallint(6) DEFAULT NULL,
  `ssid` varchar(100) COLLATE utf8_slovenian_ci DEFAULT NULL,
  PRIMARY KEY (`ix`)
) ENGINE=InnoDB  DEFAULT CHARSET=utf8
COLLATE=utf8_slovenian_ci AUTO_INCREMENT=1;
```

4.8 Razred ProbeRequest

En objekt razreda ProbeRequest predstavlja eno vrstico, zapisano v tabeli »requests« v

podatkovni bazi »wlananalysis«. Vsebuje toliko spremenljivk, kolikor ima tabela stolpcev. Objekt razreda ProbeRequest lahko kreiramo z enim izmed dveh konstruktorjev:

```
public ProbeRequest(Date date, Timestamp timestamp, String
sourceMAC, int ssi, String ssid)
```

```
public ProbeRequest(long index, Date date, Timestamp
timestamp, String sourceMAC, int ssi, String ssid)
```

Razlikujeta se v številu parametrov, ki jih podamo – drugi ima, poleg tistih parametrov, ki jih vsebuje prvi, dodan še parameter index tipa long in se ga uporabi pri kreiranju objekta, ko ga beremo iz baze. Ob pisanju v bazo parameter index ni pomemben, saj se avtomatsko povečuje, ker ima v strukturi dodano opcijo »AUTO_INCREMENT«.

Poleg tako imenovanih »get« funkcij za vsako izmed lokalnih spremenljivk je v razredu prisotna še redefinicija funkcije »equals« za primerjavo dveh objektov razreda:

```
public boolean equals(ProbeRequest probe)
{
    if (probe == null)
        return false;
    else if (probe == this)
        return true;

    Date probeDate = probe.getDate();
    Timestamp probeTimestamp = probe.getTimestamp();
    String probeSourceMAC = probe.getSourceMAC();
    String probeSSid = probe.getSSid();

    if (probeDate.compareTo(this.date) == 0 &&
         probeTimestamp.compareTo(this.timestamp) == 0 &&
         probeSourceMAC.equals(this.sourceMAC) &&
         probeSSid.equals(this.ssid))
        return true;
    else
        return false;
}
```

Dva objekta ProbeRequest sta enaka, če so enake spremenljivke:

- date,
- timestamp,
- sourceMAC,

- SSID.

Spremenljivka SSI se tu ne upošteva.

4.9 Branje konvertiranih podatkov iz .parsed datotek

Pri pretvorbi podatkov v koraku 4.6 smo iz celotnih okvirjev dobili samo tiste podatke, ki nas zanimajo. Ustvarili smo toliko ».parsed« datotek, kolikor je datotek ».cap« v vhodnem direktoriju – tega na začetku s pomočjo datotečnega dialoga (FileDialog) poiščemo na disku. Vsaka datoteka ».parsed« pa ima toliko vrstic, kolikor okvirjev je program tcpdump zajel in zapisal. Vsaka vrstica ima naslednjo obliko:

```
2013-06-28||12:01:12.471934||-58dB||SA:b1:c3:f5:cd:71:g3||(LT)
```

Funkcija »parseSourceRequests« v razredu Data ima naslednjo glavo:

```
public static ArrayList<ProbeRequest>
parseSourceRequests(String sourceDirectory, ProbeRequest
lastProbeRequest)
```

Najprej pridobimo iz baze podatke o okvirju, ki smo ga po času z zajemom sprejeli kot zadnjega. Nato iz tega okvirja pridobimo njegov datum. Ko pridobimo datum, temu odštejemo en dan, saj so v večini primerov v eni ».cap« datoteki shranjeni okvirji iz dveh dni. Primer:

Datoteka output2014-06-28.cap

```
2013-06-28||23:58:45.218451||-48dB||SA:b1:c3:f5:cd:71:g3||(LT)
2013-06-28||23:59:58.471934||-56dB||SA:a1:c4:ea:67:f4:4e||(LT)
2013-06-29||00:00:12.148931||-59dB||SA:c1:ab:2f:5a:9b:cc||(LT)
```

Tako je v tem primeru zadnji datum 2013-06-29 znotraj datoteke, ki ima v imenu datum 2013-06-28. Preveriti je potrebno, če je v tej datoteki prišlo do sprememb (dodani novi okvirji s poznejšim časom).

```
Date fileDate, lastDate;
SimpleDateFormat sfd = new SimpleDateFormat("yyyy-MM-dd");
Calendar c = Calendar.getInstance();
if (lastProbeRequest != null) //pridobimo zadnji ProbeRequest
     c.setTime(sfd.parse
               (lastProbeRequest.getDate().toString()));
else //če ni ničesar v bazi, moramo brati vse datoteke
     c.setTime(sfd.parse("1999-12-12"));
```

```
c.add(Calendar.DATE, -1);
lastDate = new Date(c.getTimeInMillis());

for (File file : listOfFiles)
     if (file.isFile() &&
FilenameUtils.getExtension(file.getName()).equals("parsed")) {
     fileDate = Date.valueOf //pridobimo datum iz imena
     (FilenameUtils.getBaseName(file.getName()).substring(6));
if (fileDate.compareTo(lastDate)==0 ||
fileDate.compareTo(lastDate) > 0)
//če je datum enak ali večji kot zadnji vnešeni
     listOfParsedFiles.add(file);} //ga dodamo v vrsto
```

Nato je potrebno vsako datoteko iz seznama »listOfParsedFiles« brati po vrsticah in posamezne vrstice razdeliti glede na delilec, ki ga skupaj tvorita dva znaka: »||«. Če je število polj, ki jih dobimo po razdelitvi, enako pet, potem ima okvir vse podatke in ga vnesemo na seznam za zapisovanje v podatkovno bazo.

4.10 Filtriranje okvirjev poskus zahteve

V zgornjem koraku smo filtrirali tiste okvirje, ki so že zapisani v naši podatkovni bazi. V nadaljevanju bomo izvedli še dodatno filtriranje.

Oddajanje okvirjev poskus zahteve je lahko zelo hitro. Velikokrat naprava odda več teh okvirjev znotraj ene sekunde. Primer takšnega oddajanja in zajema:

```
2013-06-28||12:01:12.450163||-57dB||SA:e0:63:e5:cd:77:d8||(LT)
2013-06-28||12:01:12.452553||-56dB||SA:e0:63:e5:cd:77:d8||()
2013-06-28||12:01:12.471934||-58dB||SA:e0:63:e5:cd:77:d8||(LT)
2013-06-28||12:01:12.472949||-59dB||SA:e0:63:e5:cd:77:d8||()
```

Kot vidimo, je med izpisanimi štirimi okvirji zelo malo razlik. Razlikujejo se po času v nanosekundah in po SSID polju – ta se sicer tudi ponavlja. SSI polja tu ne upoštevamo. Odločili smo se, da takšne natančnosti pri času ne potrebujemo, zato smo odstranili del časa, v katerem so zapisane nanosekunde, in tako so vsi zgoraj izpisani okvirji dobili čas: 12:01:12.000000. Sedaj vidimo, da imamo dejansko podvojene okvirje (zopet SSI polja ne upoštevamo) in da bi bil zapis le-teh v podatkovno bazo redundanten. V naslednjem koraku take okvirje filtriramo še po polju SSID in jih dodajamo na seznam enoličnih okvirjev znotraj enakega časa, ki ga, ko nastopi okvir z drugačnim časom, dodamo na celoten seznam okvirjev za vpis v podatkovno bazo.

```
//za vsako drugačno sekundo seznam enoličnih okvirjev
if (timestamp.equals(previousTimestamp))
{
     boolean found = false;
```

```
     for(ProbeRequest pr : differentPrWithin1Second)
     //trenutni okvir primerjamo z vsakim v seznamu enoličnih
okvirjev
          if(pr.equals(currentPR))
              found = true; //če je že na seznamu
          if (!found) //če ni, ga v seznam dodamo
              differentPrWithin1Second.add(currentPR);
}
else //če je čas že drugačen
{
    probeRequestList.addAll(differentPrWithin1Second);
    //dodamo vse okvirje iz trenutnega seznama enoličnih
    differentPrWithin1Second.clear();
    //seznam počistimo
    differentPrWithin1Second.add(currentPR);
    //in v »novega«, dodamo trenutni ProbeRequest
}
```

4.11 Vpisovanje v podatkovno bazo

V prejšnjem koraku smo ustvarili seznam objektov ProbeRequest, katere moramo sedaj zapisati v bazo. V pomoč so nam parametrizirane izjave [18] (prepared statement, tudi parameterized statement). To so optimizirani in tipizirani SQL stavki, ki se uporabljajo ob zaporednem izvajanju enakih ali podobnih SQL stavkov, predvsem ob vpisovanju in posodabljanju (INSERT, UPDATE stavki) podatkov v podatkovni bazi. Izvajanje parametrizirane izjave poteka v treh korakih:

1. parametrizirani izjavi podamo njeno obliko, v kateri dejanske vrednosti parametrov nadomestimo z znakom »?«,
2. sistem za upravljanje podatkovnih baz (DBMS; **D**ata**b**ase **M**anagement **S**ystem) izjavo optimizira, prevede in jo shrani,
3. aplikacija parametrizirani izjavi poda dejanske vrednosti parametrov, sistem za upravljanje podatkovnih pa izjavo izvede.

Z razredi iz paketa java.sql ustvarimo parametre za zapis v pravilni obliki.

```
ArrayList<String> columnNamesList = getTableColumnNames(conn);
//pridobimo imena stolpcev v tabeli »requests«
if (columnNamesList.size() != 5) //index polja ne štejemo
    throw new Exception("Table error!")
String insertIntoSQL = "INSERT INTO "+tblName+"(";
//sestavimo obliko parametrizirane izjave
for (String columnName : columnNamesList)
     insertIntoSQL += columnName + ",";
insertIntoSQL = insertIntoSQL.substring(0,
```

```
insertIntoSQL.length()-1) + ") VALUES ("; //izbrišemo zadnjo ,
for (String columnName : columnNamesList)
     insertIntoSQL += "?,";
insertIntoSQL = insertIntoSQL.substring(0,
insertIntoSQL.length()-1) + ");"; //izbrišemo zadnjo ,
//INSERT INTO requests (date, timestamp, source, ssi, ssid)
VALUES (?, ?, ?, ?, ?);

PreparedStatement st = conn.prepareStatement(insertIntoSQL);
ResultSet rs;
for (ProbeRequest probeRequest : probeRequestsList)
{
     //za vsak objekt ProbeRequest v seznamu
    st.setDate(1,probeRequest.getDate());
    st.setTimestamp(2, probeRequest.getTimestamp());
    st.setString(3, probeRequest.getSourceMAC());
    st.setInt(4, probeRequest.getSsi());
    st.setString(5, probeRequest.getSSid());
    st.executeUpdate();
    //nastavi parametre in izvedi izjavo
}
```

4.12 Generiranje diagramov s knjižnico JfreeChart

Delo s knjižnico JFreeChart je zaradi dobre dokumentacije, primerov in razširjenosti (za veliko bazo uporabnikov je na voljo tudi veliko primerov [19]) zelo enostavno. Ko kreiramo nov diagram, moramo povedati, za kakšen diagram gre, mu podati podatke, nad katerimi knjižnica diagram ustvari, in podati posamezne oznake, ki bodo diagramu dale pomen.

Ustvarili smo posebne razrede za različne tipe grafov, ki so uporabljeni v naši rešitvi. Posamezen objekt razreda ustvari nov diagram, ga po naših potrebah uredi in prikaže. Konstruktor razreda BarChart (diagram se odpre v novem Jdialog objektu znotraj očeta, ki je tudi tipa JDialog):

```
public BarChart(JDialog parent, final String title,
CategoryDataset dataset, String xLabel, String yLabel)
{
    JFreeChart chart=createChart(dataset,title,xLabel,yLabel);
    //ustvarimo diagram
    chart.removeLegend();
    ChartPanel chartPanel = new ChartPanel(chart);
    JDialog chartDialog = new JDialog(parent);
    chartDialog.setModal(true);
    //dokler okna ne zapremo, ta ne izgubi fokusa
    chartDialog.add(chartPanel);
    chartDialog.setLayout(new GridLayout(1, 1));
```

```
    chartDialog.setLocation(parent.getX()+100,
                            parent.getY()+100);
    chartDialog.setSize(800, 400);
    chartDialog.setVisible(true);
}

private JFreeChart createChart(CategoryDataset dataset, String
chartName, String xLabel, String yLabel)
{
    JFreeChart chart = ChartFactory.createBarChart(
        chartName,
        xLabel,
        yLabel,
        dataset,
        PlotOrientation.VERTICAL,
        true,
        true,
        false);
    return chart;
}
```

4.13 Popravljanje funkcije za opis diagramov

V programski rešitvi se v nekaterih analizah pojavljajo diagrami, ki prikazujejo časovna obdobja. Za prikaz diagrama je potrebno to časovno obdobje spremeniti v vrednost, ki jo lahko knjižnica JFreeChart prikaže – številke (tipi double, int). V neki analizi smo časovna obdobja za posamezen MAC naslov iz podatkovne baze pridobili na sledeč način:

Psevdokoda:

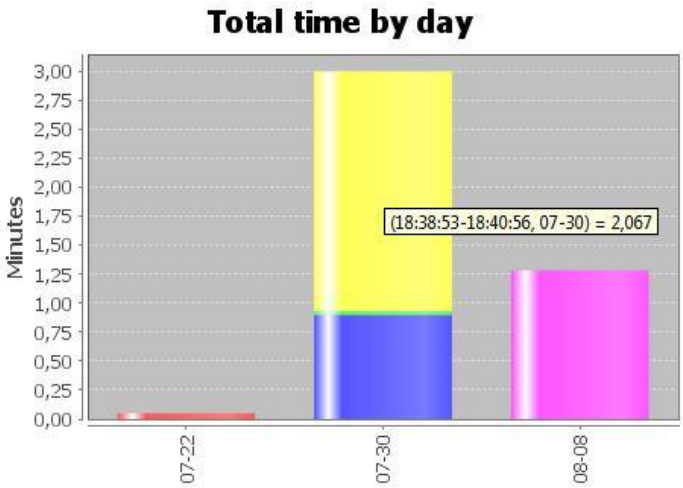
```
občutljivost = 120 sekund;
seznamIntervalov = nov seznam;
interval = 0 sekund;

if (ProbeRequestList nima nobenega ali samo en objekt)
     zaključi //ni nobenega intervala
prviTimestamp = parameter timestamp prvega objekta v seznamu

for (od drugega objekta v ProbeRequestList dalje)
{
     drugiTimestamp = parameter timestamp objekta ProbeRequest
     intervalMedObjektoma = drugiTimstamp – prviTimestamp
     if (intervalMedObjektoma < občutljivost)
          interval += intervalMedObjektoma
     else
     {
          dodamo interval na seznamIntervalov
```

```
            interval nastavimo nazaj na 0
        }
}
```

Tako smo intervale pridobili v sekundah. Odločili smo se, da v diagramih intervalov ne bomo prikazovali v sekundah, temveč v minutah. Tako pridobljene intervale smo delili s 60 in vrednosti predali knjižnici za generiranje diagramov. Diagrami so delno interaktivni – če se z miškinim kazalcem ustavimo na posameznem delu diagrama, se nam bo prikazal opis, ki bo vseboval x in y vrednosti komponente izrisanega diagrama.



Slika 15: Primer opisa vrednosti na grafu

Iz zgornjega primera je razvidno, da je bila naprava v dosegu usmerjevalnika na datum 07-30 od 18:38:53 do18:40:56 – interval je tako prikazan kot decimalna vrednost: 2,067. Opis smo popravili tako, da se je interval prikazoval kot časovna vrednost. Potrebno je bilo redefinirati funkcijo »generateToolTip«.

```
renderer.setBaseToolTipGenerator(new
StandardCategoryToolTipGenerator() {
@Override
    public String generateToolTip(CategoryDataset ds, int row,
                                                    int column){
        String time = ds.getRowKey(row).toString();
        //vrednost zmnožimo s 60 da dobimo sekunde
        double value = (double)ds.getValue(row, column);
        double valueInSeconds = value * 60;
        int minutes = (int)valueInSeconds / 60;
        int hours = minutes / 60;
        if (hours > 0)
            minutes = minutes % 60;
        int seconds = (int)valueInSeconds % 60;
```

```
        return "( "+time+" )" + " - " + (hours > 0 ?
                String.format("%02d", hours)+":" : "")+
                String.format("%02d", minutes) + ":" +
                String.format("%02d", seconds);
        //formatiramo izpis }
        });
```

Poglavje 5

Testiranje in rezultati

Ko smo uspeli nastaviti usmerjevalnik tako, da je zajemal podatke, smo ga najprej postavili za testno obdobje enega tedna na isto lokacijo, kot smo jo uporabili v nadaljevanju. Po pregledu rezultatov poskusnega tedna smo se odločili, da usmerjevalnik pustimo zajemati kar na tej lokaciji, saj bomo tako mogoče izvedeli zanimive informacije o zaposlenih na fakulteti.

5.1 Vzpostavitev testnega okolja

Kot lokacijo za zajem podatkov smo izbrali parkirišče za stavbo Fakultete za računalništvo in informatiko. Usmerjevalnik je bil postavljen na okensko polico Laboratorija za informatiko in imel tako dober doseg po večini parkirišča.

Ker usmerjevalnik za zajem želenih podatkov ne potrebuje internetne povezave, je nismo priključili. Tako konfiguriran usmerjevalnik lahko postavimo kamor koli, kjer imamo možnost priklopa električne energije. Potrebno je omeniti še to, da usmerjevalnik privzeto pridobi datum in čas preko internetne povezave – v nastavitvah DD-WRT spletnega vmesnika se lahko določi naslov časovnih strežnikov, s katerimi se ob zagonu usmerjevalnik skuša povezati in tako prejme točen datum in čas. V primeru, da se s temi strežniki ne more povezati, se nastavi privzeti čas 1. 1. 1970, 00:00:00 (epoch čas [20]). Ker možnosti internetne povezave na usmerjevalniku nimamo, lahko datum in čas določimo na drugačen način. V terminalu obstaja ukaz »date«, s katerim se izpiše trenutni datum. V skripti, ki se izvede ob zagonu usmerjevalnika, smo zato dodali naslednjo vrstico:

```
date –s 1306280945
```

Stikalo –s nastavi datum v pripadajočem argumentu, ki mora imeti obliko YYMMDDHHMM. Zgornji ukaz je nastavil datum in čas na: 09:45, 28. junij 2013. Na tak način smo 28. junija spremenili skripto, jo shranili, usmerjevalnik ugasnili in ga ponovno vklopili, ko je ena izmed naših ur (na telefonu) pokazala 09:45. Tako smo na usmerjevalniku nastavili čas, ki se je le za nekaj sekund razlikoval od dejanskega.

Težava se pojavi v primeru, če med zajemom zmanjka električne energije. Ker brez pomoči zunanjega vpliva (interneta) ni možno določiti, koliko časa ni bilo električne energije, bi se čas usmerjevalnika po vnovičnem zagonu zopet postavil na tisto vrednost, katera je vpisana v skripti, ki se izvede ob zagonu. Tako bi usmerjevalnik začel prepisovati z zajemom že ustvarjene datoteke.

Zajem podatkov je trajal od 28. 06. do 14. 08. 2013.

5.2 Tipi analiz, podprti v aplikaciji

Aplikacija za analizo zajetih okvirjev Probe Request omogoča različne vrste analiz. Razvrstili smo jih v 3 glavne skupine:

- časovna analiza,
- analiza MAC naslovov,
- analiza SSID polja.

5.2.1 Časovne analize

Na maski za časovno analizo imamo na voljo več različnih tipov obdobij. Ta so razvrščena po dnevih, tednih in mesecih (v primeru, da bi podatke želeli zajemati več let, bi bilo smiselno aplikacijo razširiti tako, da bi bilo mogoče izbrati podatke za prikaz po letih). V vsakem izmed seznamov lahko izberemo eno obdobje ali več (enojni interval).

Vsak izmed treh seznamov ima dve možnosti analize: lahko se analizira posamezno obdobje (izbor enega dneva, tedna, meseca) ali pa se izbere interval obdobij. Glede na izbor se spreminja izrisani diagram. V izbranem obdobju se analizira število unikatnih naslovov MAC (število različnih naprav). Analizo sprožimo s klikom na gumb, ki pripada želenemu obdobju:

- dnevi
  - izbira enega dneva: diagram števila naprav po urah glede na izbrani dan,
  - izbira več dni : diagram števila naprav na izbrane dni,
- tedni
  - izbira enega tedna: diagram števila naprav po dneh v izbranem tednu,
  - izbira več tednov: diagram števila naprav po izbranih tednih,
- meseci
  - izbira enega meseca: diagram števila naprav po tednih v izbranem mesecu,
  - izbira več mesecev: diagram števila naprav v izbranih mesecih.

5.2.2 Analize MAC naslovov

Na levi strani maske za analizo MAC naslovov [22] se nam napolni seznam vseh različnih MAC naslovov, ki so bili zabeleženi kadar koli v času zajema. Njihovo število je zapisano v oznaki pod seznamom. Vsak zapis ima obliko:

MAC_naslov-številka

Številka nam pove, ob koliko različnih dnevih v celotnem časovnem obdobju zajema se je MAC_naslov pojavil v območju našega usmerjevalnika. Celoten seznam je nato urejen ravno po tej vrednosti – od tistega, ki je bil v območju največkrat, do tistih, ki so bili v območju le enkrat.

Celoten seznam lahko filtriramo tudi po proizvajalcu naprave. Da je ta opcija mogoča, je pred zagonom analize MAC naslovov potrebno izbrati vhodni direktorij, ki mora vsebovati direktorij z imenom »manufacturers«. Ta direktorij mora vsebovati tekstovne datoteke z imeni proizvajalcev (v našem primeru: apple, samsung, htc, nokia, sony), vsaka izmed teh datotek pa vsebuje vpise, ki predstavljajo MAC naslovni prostor posameznega proizvajalca [21]. Primer zapisov v datoteki samsung.txt:

```
00:00:F0
00:07:AB
00:12:47
```

MAC naslovni prostor zavzema 6 od 12 znakov MAC naslova. Vsaka organizacija (proizvajalec) si (načeloma) lasti več naslovnih prostorov. Ko je organizaciji naslovni prostor dodeljen, lahko ta izdela naprave z MAC naslovi:

- od: naslovni_prostor:00:00:00
- do: naslovni_prostor:FF:FF:FF

V zgornjem primeru lahko rečemo, da MAC naslov 00:00:F0:00:00:01 pripada Samsungu.

Ko izberemo želenega proizvajalca, s klikom na »Update list!« posodobimo seznam MAC naslovov tako, da bodo našteti samo tisti, ki pripadajo izbranemu proizvajalcu. Pod seznamom se bo število različnih naslovov primerno posodobilo.

V seznamu lahko posamezno napravo (MAC naslov) tudi izberemo in podrobneje analiziramo. Z izbiro naprave in klikom na gumb »Analyze MAC!« se na desni strani maske:

- v zgornji seznam izpišejo dnevi, na katere je bila naprava v območju usmerjevalnika,
- pojavi večstolpični diagram – vsak stolpec prikazuje, koliko časa je bila izbrana naprava na posamezne dneve v območju usmerjevalnika,
- s klikom na posamezen dan v seznamu dni v spodnji seznam izpišejo ure v tistem dnevu, v katerih je bila naprava v območju usmerjevalnika.

Ob premiku kurzorja na katerega izmed intervalov v večstolpičnem diagramu se pojavi opis, v katerem so zapisani dolžina ter začetni in končni čas intervala.

5.2.3 Analize imen brezžičnih omrežij

Na maski za analizo imen brezžičnih omrežij (SSID) sta dva seznama. Levi se napolni z vsemi različnimi imeni brezžičnih omrežij, njihovo število pa je zapisano v oznaki pod seznamom. Zraven je še zapisano število tistih okvirjev, v katerih se je SSID polje med zajemom (zapisom) poškodovalo. Vzrok bi lahko bil hrošč v programu tcpdump [23], lahko pa, da je do napake prišlo že pri oddaji okvirja od naprave ali pri sprejemu na usmerjevalniku. Če datoteko z zajemom odpremo v programu Wireshark in poiščemo tak okvir, bomo v predelu, kjer bi se moralo izpisati polje SSID, videli le neberljive znake. Vsak zapis ima obliko:

Ime_brezžičnega_omrežja – številka

Številka nam pove, koliko različnih naprav (MAC naslovov) pošilja Probe Request okvirje, ki imajo vrednost polja SSID nastavljeno na Ime_brezžičnega_omrežja. Celoten seznam je nato urejen ravno po tej vrednosti – od tistega, ki se pojavlja na največ različnih napravah, do tistih, ki se pojavijo le enkrat.

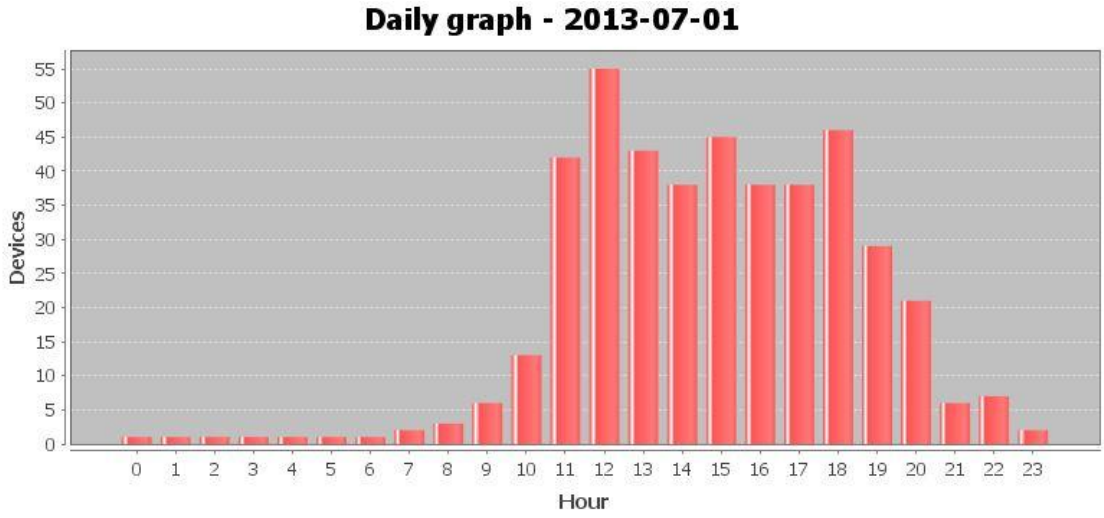
Seznam ima dve možnosti analize. Če izberemo eno ime brezžičnega omrežja in kliknemo na gumb »Analyze SSID«, se bodo v desnem seznamu izpisale vse naprave, ki oddajajo okvirje poskus zahteve brezžičnemu omrežju s tem imenom. Če pa izberemo dve ali več imen brezžičnih omrežij in kliknemo na gumb »Analyze SSID«, se nam prikaže presek tistih naprav, ki oddajajo okvirje poskus zahteve na točno ti dve ali več različnih izbranih imen brezžičnih omrežij.

5.3 Primeri analiz

V nadaljevanju so izrisani primeri analiz, ustvarjenih z našo aplikacijo. Predvideli bomo, da vsak unikaten MAC naslov predstavlja eno osebo – če upoštevamo lokacijo postavitve, lahko sklepamo, da gre v večini za zaposlene ter študente.

5.3.1 Število različnih MAC naslovov v danem obdobju

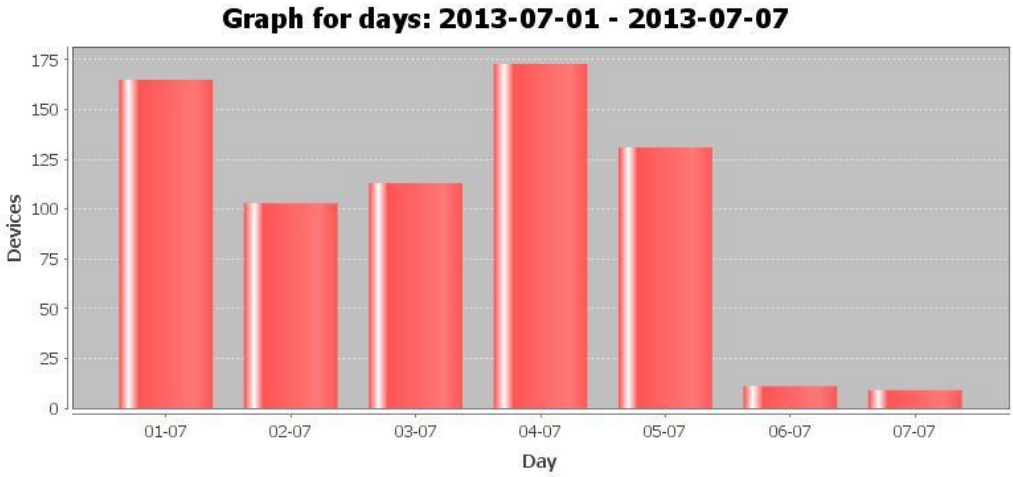
5.3.1.1 Urna analiza na izbrani dan



Slika 16: Urna analiza

V seznamu dni smo izbrali ponedeljek, 1. 7. 2013. Od polnoči do okrog 7. ure zjutraj je bila v območju le ena naprava, za katero bi, če bi se vzorec ponavljal na druge dni, lahko dejali, da je statična in vedno vklopljena. Po 10.00 oz. okrog 11. ure se je v bližini usmerjevalnika zvrstilo največ zaposlenih – prihod na delo. Vidimo tudi, da je število začelo upadati med 19. in 21. uro. Iz tega bi lahko sklepali, da se delovni dan zaposlenih na fakulteti poleti prične okrog 10.30, konča pa blizu 19. ure. Da bi lahko to izjavo tudi potrdili, bi morali primerjati vzorce še ob preostalih dnevih poletja, ko se izpitno obdobje zaključuje.

5.3.1.2 V več izbranih dnevih



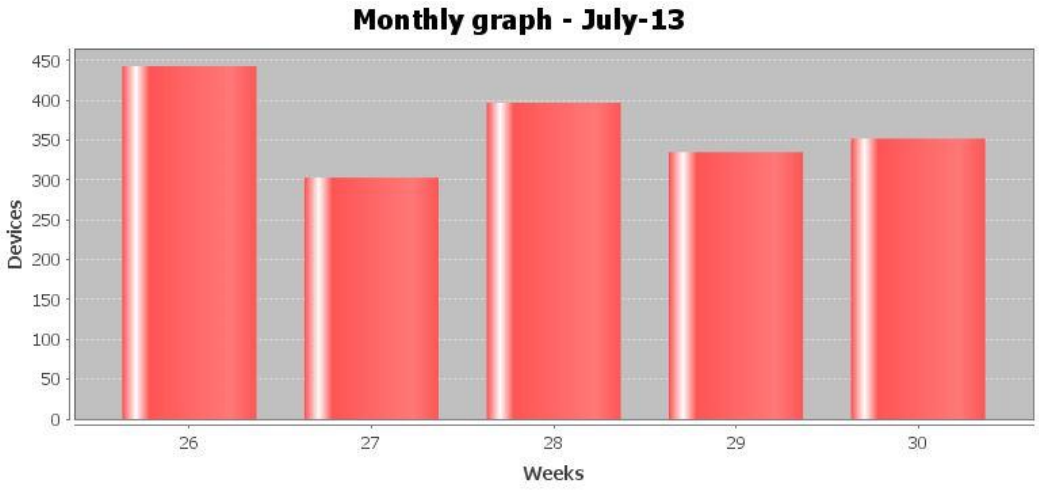
Slika 17: Večdnevna analiza

V seznamu dni smo izbrali sedem dni: od 1. 7. do 7. 7. 2013. Zelo dobro se vidi, da sta zadnja dva dneva v tednu sobota in nedelja, ko je število oseb drastično nižje od preostalih dni. Za

torek in sredo, ko je bilo število oseb nižje za več kot četrtino kot v ponedeljek ali četrtek (tudi petek), bi lahko sklepali, da na ta dva dneva ni bilo nobenega izpita – v začetku julija izpitno obdobje namreč še vedno traja.

Enak diagram bi lahko izpisali, če bi v seznamu tednov izbrali 27. teden.

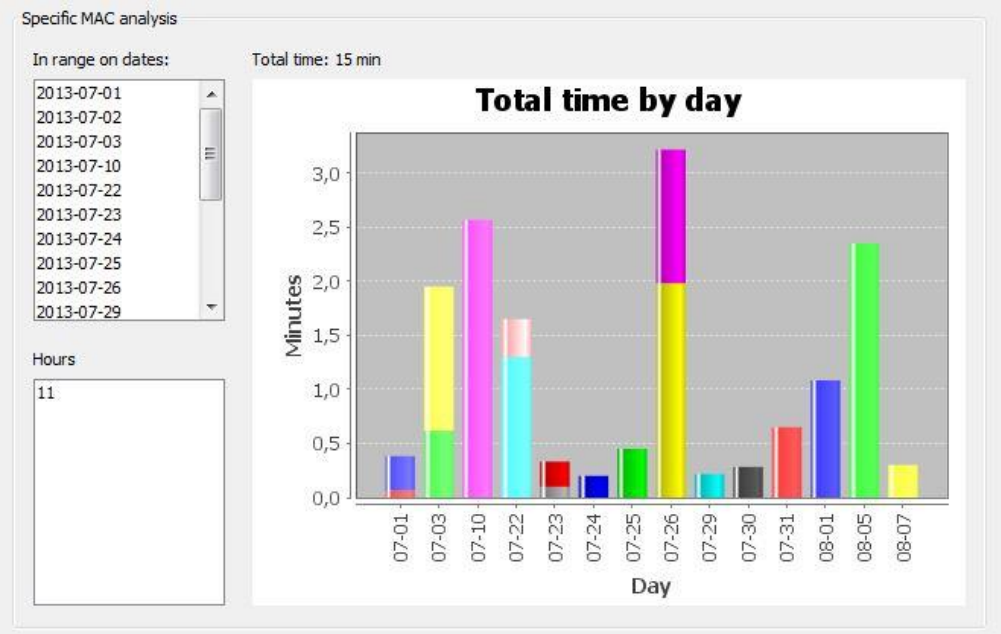
5.3.1.3 V izbranem mesecu



Slika 18: Mesečna analiza

V seznamu mesecev smo izbrali julij 2013. Diagram kaže število različnih oseb po tednih. Opozoriti je potrebno tudi na to, da je zadnji, 30. teden, že del meseca avgusta in se to tudi upošteva v analizi (upošteva se celoten teden, ne samo dnevi, ki pripadajo mesecu juliju). Zaradi prekratkega obdobja zajemanja podatkov (in ker nimamo vzorcev od prej) na mesečni ravni zaenkrat ne moremo ničesar sklepati.

5.3.2 Časovna analiza posameznega MAC naslova



Slika 19: Časovna analiza MAC naslova

Pri analizi te specifične naprave (oz. osebe) smo lahko opazili zanimive stvari. Ko smo v posameznih datumih pregledali, ob katerih urah je bila naprava v območju usmerjevalnika, smo ugotovili, da se ponavljajo le prve in zadnje ure v dnevu. Največkrat se je naprava prvič pojavila med 11. in 12. uro, zadnjič pa med 17. in 18. uro. Predvidevamo lahko, da ima oseba na napravi brezžično omrežje vedno vklopljeno. Dopoldne je prišla na delo, popoldne odšla (ponavljajoči se vzorci), medtem pa ni bilo znakov aktivnosti (skupni čas v dosegu vseh dni je 15 minut, kar bi pomenilo, da je bila naprava v dosegu le takrat, ko je bila oseba na parkirišču - potem je odšla proti svojemu delovnemu mestu), iz česar lahko sklepamo, da je oseba zaposlena na delovnem mestu, ki ni na tej strani fakultete.

5.3.3 Iskanje MAC naslovov po proizvajalcu

Izbrali smo analizo MAC naslovov in filtracijo po proizvajalcu. Izbrali smo proizvajalca Apple. Od 1459 različnih naprav jih je 473 tega proizvajalca. Samo iz tega podatka ne moremo sklepati ničesar, razen ugotovitve, da je skoraj 1/3 naprav, ki smo jih zaznali, Applovih, vendar je velika večina takšnih, ki so se v območju pojavile le enkrat. Če bi želeli pridobiti informacijo o napravah zaposlenih, bi morali uvesti še dodaten filter, kot na primer: dodatno filtriraj le tiste naprave, ki se v območju pojavijo vsaj desetkrat na mesec.

5.3.4 Iskanje MAC naslovov, ki iščejo dva ali več različnih SSID

Izbrali smo analizo SSID polj in iz seznama vseh brezžičnih omrežij izbrali dve: »eduroam« (študentsko omrežje na fakultetah) in »LPT« (brezžično omrežje v Ljubljani). Presek teh nam je pokazal, da je 228 naprav, ki imajo shranjeno brezžično omrežje »eduroam«, ter 21 naprav, ki imajo shranjeno brezžično omrežje »LPT«, od teh pa jih ima kar 12 shranjenih obe brezžični omrežji. Če ima nekdo shranjeno omrežje »LPT«, lahko sklepamo, da se prijavlja v brezžično omrežje na različnih lokacijah po Ljubljani. Če ima ista naprava shranjeno še omrežje »eduroam«, za katero vemo, da je študentsko oz. akademsko omrežje, pa lahko morebiti sklepamo, da je lastnik takšne naprave študent.

5.4 Ugotovitve

Že pri nekaterih zgoraj navedenih poskusih smo prišli do zanimivih ugotovitev. Za podrobnejšo in bolj kompletno analizo pa bi bili potrebni vsaj trije dodatni pogoji:

- daljše časovno obdobje zajemanja podatkov (če bi podaljšali čas zajema na celoten avgust in september, bi morda lahko opazili drastičen skok v številu novih oz. različnih naprav),
- bolj optimalna izbira lokacije (v našem primeru - nekje bližje sredini parkirišča),
- dodati nove funkcije aplikaciji za analizo.

Poglavje 6

Poslovne priložnosti in pravni vidiki sledenja uporabnikom

V prejšnjem poglavju smo si ogledali nekaj primerov, kaj vse je mogoče ugotoviti iz tako »majhnega« podatka, kot je okvir zahteve. Za marsikoga nič posebej zanimivega, a zelo zanimivo in mamljivo predvsem za – trgovce. Kot velja za vsako takšno zbiranje (osebnih) podatkov, tudi tu obstajajo pravni vidiki in omejitve, mimo katerih ne moremo.

6.1 Poslovne priložnosti

Trgovci iščejo vsak možen način za poskus povečanja prodaje. Podatki o naših (nakupovalnih) navadah se zbirajo vsak dan: vsakič, ko v trgovini Spar uporabimo Spar plus kartico [24], vsakič, ko v Mercatorju uporabimo njihovo Piko [25], medtem ko smo prijavljeni v Googlov račun in vnesemo iskalni niz v iskalnik Google [26] itd. Nekateri večji trgovci, ki tako imenovanih kartic zvestobe (še) nimajo, pa najemajo posebej za to specializirane agencije, ki za njih takšne analize opravljajo. Z uporabo teh storitev uporabniki avtomatsko soglašajo z zbiranjem takšnih podatkov za namene: ciljnega in personaliziranega oglaševanja na podlagi proučevanja nakupovalnih navad, ter, posledično, pospeševanja in povečanja prodaje teh podjetij.

Sliši se »zlobno«, a na drugi strani so uporabniki tisti, ki takšno početje dovoljujejo. Vsi ti sistemi delujejo po principu zavednega odločanja posameznika (opt-in), da dovoljuje in soglaša z zbiranjem svojih osebnih podatkov. Kaj je torej cilj takšnega početja za uporabnika? Zakaj se ljudje odločajo posredovati svoje podatke za obdelavo? Prilagajanje uporabniku (potrošniku) ni nujno slaba stvar in v večini primerov sta na boljšem oba – tako trgovec kot tudi potrošnik. Potrošnik zato, ker mu je trgovec priskrbel dobrine, ki jih on dejansko potrebuje, trgovec pa zato, ker je lahko ustregel njegovim potrebam - na podlagi raziskave nakupovalnih navad je dobrine priskrbel in jih prodal.

S sistemom oz. aplikacijo, ki je bila predstavljena v sklopu tega diplomskega dela, bi si pri optimizaciji poslovanja zelo pomagali v različnih segmentih trga in tudi v organizacijah.

6.1.1 Tehnična trgovina

Z zajetjem podatkov v trgovini s tehničnim blagom bi si lahko trgovec zelo pomagal pri prodaji. Ugotovil bi:

- ob katerih urah je največ obiskovalcev,
- ali je na podlagi tega smiselno koga zaposliti/odpustiti,

- kako se poveča obisk, kadar poteka akcijska prodaja,
- ali se kupci vračajo in je smiselno uvesti program zvestobe,
- kako prilagoditi ponudbo (na podlagi analize proizvajalcev naprav uporabnikov),
- če imamo v sistemu še opcijo določanja lokacije, bi dobil pregled, v katerem delu trgovine se uporabniki najbolj zadržujejo, in bi lahko tam okrepil ponudbo.

6.1.2 Fakulteta

Tudi fakulteta bi lahko optimizirala del svojega delovanja na podlagi zbranih informacij o prihodu zaposlenih na parkirišče in njihovem odhodu:

- kakšen je povprečni delovni čas (glede na letni čas),
- na podlagi tega lahko premislimo, ali bi bilo smiselno parkirišče osvetliti (če osvetlitev že obstaja, jo lahko prilagodimo glede na zbrane informacije),
- podobno lahko storimo z ogrevanjem notranjih prostorov (bolj optimalno izkoriščanje).

6.2 Pravni vidiki sledenja uporabnikom

V nekaterih trgovinah in organizacijah po svetu podobne rešitve že uporabljajo v namene tržnih raziskav [30]. V ZDA, kjer so takšne in podobne metode profiliranja strank trenutno najbolj razširjene, tovrstno zbiranje podatkov (zaenkrat) ni nedovoljeno. Kljub temu gre za veliko količino podatkov, nekatere izmed njih bi lahko šteli tudi kot osebne (MAC naslov), kar lahko s primerno obdelavo, kombinirano z drugimi sistemi za nadzor v trgovini (video nadzor), predstavlja resno grožnjo varovanju zasebnosti. Trgovci sicer pri takšnem zbiranju podatkov pravijo, da:

- zbrani podatki niso nič drugačni od tistih, ki jih uporabniki spletnih nakupovalnih strani puščajo na teh straneh,
- ko nakupujete, ste v javnosti - in v javnosti se ne pričakuje zasebnosti.

Težavo predstavlja tudi dejstvo, da je večina tistih uporabnikov, ki imajo brezžične storitve vedno vklopljene in tako na vsakem koraku oddajajo svoje podatke, nepodučenih oziroma premalo osveščenih o uporabi in nevarnostih brezžičnih omrežij. Ko so v neki trgovini, ki je izvajala takšno zbiranje podatkov, zagovorniki zasebnosti dosegli, da se uporabnike pred vhodom v trgovino z izobešenim znakom obvesti o programu zbiranja podatkov, so se začeli zaradi tega pritoževati. Na podlagi tega lahko tudi sklepamo, da je marsikdo spremenil svoje navade, kar se tiče brezžičnih storitev.

Številni uporabniki pa ne vidijo težav v takem početju – ko bi tak uporabnik vstopil v trgovino, bi mu ta na njegovo napravo poslala njemu prilagojen seznam artiklov, baziran na

preteklih nakupih in obiskih, iz katerih lahko izvedo, na katerih oddelkih se najdlje zadržuje (nakupovalne navade).

Pri nas se MAC naslov naprave že upošteva kot osebni podatek [31]:

»*Osebne podatke ... predstavljajo predvsem podatki o MAC naslovu (mobilne naprave) ... . MAC naslov je namreč enolično določen vsaki mobilni napravi in ga je v večini primerov težko zamenjati, s tem pa je ob upoštevanju zmožnosti tudi ostalih subjektov in običajnih vzorcih uporabe mobilnih naprav - te zelo redko posojamo drugim - treba priti do zaključka, da se omenjeni podatki nanašajo na določene ali vsaj določljive posameznike.*«

Kljub takšni definiciji MAC naslova se lahko trgovci za zajem teh in ostalih podatkov, ki so zapisani v okvirju poskus zahteve, oprejo na deseti člen Zakona o varstvu osebnih podatkov. Ta v prvem odstavku sicer pravi, da se lahko osebne podatke obdeluje le v primeru, da je bila podana osebna privolitev posameznika. V tretjem odstavku pa piše:

»Ne glede na prvi odstavek tega člena se lahko v zasebnem sektorju obdelujejo osebni podatki, če je to nujno zaradi uresničevanja zakonitih interesov zasebnega sektorja in ti interesi očitno prevladujejo nad interesi posameznika, na katerega se nanašajo osebni podatki.«

Torej v primerih, da gre za obdelavo za namene boljšega poslovanja (zakoniti interes subjekta v zasebnem sektorju), lahko sklepamo da je zajem podatkov dovoljen.

Pri našem eksperimentu se lahko opremo na sedemnajsti člen. Prvi odstavek pravi:

»Ne glede na prvotni namen zbiranja se lahko osebni podatki nadalje obdelujejo za zgodovinsko, statistično in znanstveno-raziskovalne namene.«

Obenem informacijski pooblaščenec še pove:

»Pooblaščenec je dalje mnenja, da se posameznik, ki ima omogočeno opcijo, da je njegovo mobilno napravo možno odkriti oziroma zaznati ..., zaveda oziroma bi se moral zavedati, da s tem neomejenemu oz. nedoločenemu krogu subjektov sporoča naziv svoje naprave in njen MAC naslov. Posameznik ima vselej možnost, da onemogoči zaznavanje svoje mobilne naprave ... in na ta način zavaruje svoj interes do varstva osebnih podatkov.«

Torej, uporabnik lahko največ naredi za varstvo svojih osebnih podatkov.

6.3 PayPal Beacon

V začetku septembra je podjetje PayPal na trg poslalo nov proizvod: Beacon [27]. Gre za majhno napravo v obliki USB ključka, katero kupijo trgovci, jo priklopijo na električno

omrežje in postopek z njihove strani lahko steče. Kar še potrebujejo, je, da si njihovi potrošniki na svoj pametni telefon namestijo najnovejšo PayPal aplikacijo, omogočijo Bluetooth LE (Bluetooth low-energy [28]) in (vsaj na začetku, opt-in) izrecno vklopijo beacon (del aplikacije PayPal).

Ko se bo s tako nastavljenim pametnim telefonom potrošnik približal trgovini, se bo na trgovčevem računalniku prikazala njegova slika in PayPal profil. Tako bo lahko trgovec potrošnika ob vstopu v trgovino pozdravil kar po imenu, mu ponudil njegovo najljubše naročilo, v restavraciji se bo ob približevanju gosta aktiviral signal in že se bo iskala prosta miza itd. Tudi plačevanje naj bi bilo lažje – ker bo imel trgovec na voljo profil s sliko, naj bi za plačevanje zadoščalo le ustno potrdilo, račun pa naj bi prispel preko e-pošte. *Prišel, videl, odnesel*.

PayPal podatka o prisotnosti v trgovini trgovci naj ne bi delil z drugimi. Seveda pa bo že privzeto nastavljeno prejemanje prilagojenih in posebnih ponudb posameznega trgovca in kuponov na PayPal profil, a to naj bi se dalo to izklopiti (opt-out).

Tudi PayPal tako trgovcem ponuja storitev analiziranja in proučevanja nakupovalnih navad, obenem pa potrošnikom (še bolj) poenostavlja plačevanje.

Poglavje 7

Možne izboljšave in nadgradnje

Program za analizo se da razširiti. V sklopu diplomskega dela smo pokazali le delček tistega, za kaj vse bi lahko zajem takšnih podatkov uporabili, ter le enega izmed načinov, kako do tega priti. Kot je prikazano že v marsikaterem članku, je agregacija in obdelava »surovih« podatkov tista, ki lahko:

- nekomu (osebi, ki podatke obdeluje) pridobi prednost (v informacijski dobi so informacije tiste, ki imajo visoko vrednost),
- nekoga prestraši, ujezi, začudi (oseba, ki podatke (nevede) posreduje), ko spozna, kaj vse je možno z dobro obdelavo »surovih« podatkov izluščiti.

V praksi se ta dva vidika pojavljata skupaj.

V začetku poglavja so predstavljene možne izboljšave programa za analizo ter pretvarjanja zajetih okvirjev v tekstovno obliko, v nadaljevanju se pa posvetimo SSI polju in konceptu določanja uporabnikove lokacije.

7.1 Izboljšave skripte za pretvarjanje zajetih podatkov v tekstovno obliko

Način pretvarjanja podatkov iz zajema bi se lahko spremenil. V Linuxu bi se lahko spisal program v C/C++ (isti programski jezik, v katerem sta napisana »tcpdump« in pripadajoča knjižnica »libpcap«), ki bi tekel v ozadju in skrbel za pretvorbo .cap datotek v .parsed v realnem času (ko bi tcpdump zaključil pisanje datoteke). Namesto manipulacije z nizi bi tako s pomočjo te knjižnice pretvarjali podatke na enakem nivoju in z objekti, kakršne pozna »tcpdump«.

7.2 Izboljšave programa za analizo

Zajetih podatkov je ogromno. V tabelo v bazi smo zapisali okrog 350000 vrstic probe request okvirjev – če jih ne bi filtrirali že pred zapisovanjem v bazo, bi jih bilo preko 900000 – podatke smo zajemali le malo več kot mesec in pol. Za večje in daljše zajemanje bi bilo potrebno na tem področju:

- spremeniti zasnovo tabele (da ne bi imeli vseh podatkov zapisanih v eni tabeli, saj se jih veliko ponavlja – delo z indeksi),
- nad novo zasnovo tabel uporabiti indekse,

- optimizirati poizvedbe.

Sami aplikaciji za analizo bi lahko dodali nove funkcionalnosti oziroma nove možnosti analiziranja podatkov (odvisno predvsem od tega, kakšne so naše informacijske potrebe):

- primerjava urnih analiz med različnimi dnevi,
- agregacija urnih analiz v vseh oz. izbranih dnevih (povprečni časa prihoda in odhoda),
- možnost analize MAC naslovov le tistih naprav, ki so se pojavile v danem obdobju,
- povezava analize MAC naslovov s kakšno izmed spletnih strani, kjer bi lahko pridobili informacijo o proizvajalcu danega MAC naslova,
- izris tortnega diagrama MAC naslovov po proizvajalcu,
- v primeru, da pridobimo dostop do geolokacijske baze, ki shranjuje imena brezžičnih omrežij in njihovih lokacij (Google?), lahko tako povežemo izbrane MAC naslove z lokacijami po Sloveniji (svetu).

7.3 SSI polje in koncept določanja uporabnikove lokacije s trilateracijo

Med zajemom podatkov iz okvirjev poskus zahteve smo shranjevali tudi vrednosti SSI polja. V polju SSI se shranjuje vrednost, ki predstavlja sprejeto moč signala. Izvedli smo poskus, v katerem smo dostopno točko postavili v zunanje okolje, ga nastavili tako, da je zajemanje okvirjev poskus zahteve začelo teči dve minuti po vklopu, se od dostopne točke umaknili na večjo razdaljo, nato pa se ji počasi približevali. Po pregledu rezultatov v programu Wireshark [32] smo ugotovili, da se je vrednost SSI polja s časom povečevala, vrednosti pa so sovpadale s tipično močjo sprejetega signala v napravah brezžičnih omrežij (-65 do -90 dBm [33]).

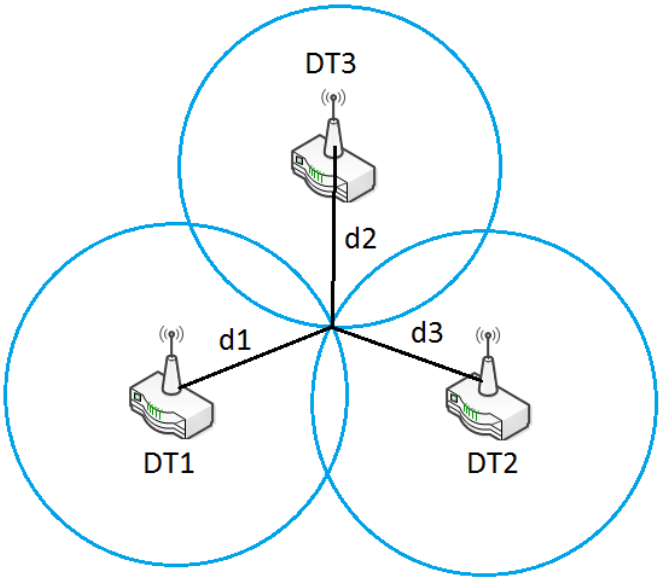
S strateško postavitvijo vsaj treh dostopnih točk bi lahko s pomočjo trilateracije določili tudi lokacijo odjemalca, ki se nahaja znotraj dometa teh dostopnih točk [34].

S pomočjo enačbe:

$d_i = p\ (\ 1 - m_i\ )$, kjer predstavlja:

- d_i izračunano razdaljo,
- p maksimalen doseg dostopne točke (v metrih),
- m_i moč sprejetega signala, izraženega v odstotkih, izračunanega kot kvocient sprejete moči signala z maksimalno možno močjo dostopne točke,
- i zaporedno število dostopne točke (1, 2, 3),

bi tako določili razdalje odjemalca od posamezne dostopne točke. Nato poiščemo presečišče vseh treh navideznih krožnic, ki jih okoli posamezne dostopne točke tvorijo izračunane vrednosti razdalj iz prejšnjega koraka.



Slika 20: Koncept določanja pozicije s pomočjo trilateracije

Zaključek

V diplomskem delu smo bralcu želeli predstaviti način, kako zajeti in prebrati podatke, ki se vsakodnevno gibljejo okoli nas. Obenem smo tudi podrobneje predstavili način delovanja brezžičnih omrežij – predvsem začetnega dela pri vzpostavljanju povezave med odjemalcem in usmerjevalnikom in marsikdo bi si verjetno predstavljal, da se o njem ne more izvedeti ničesar, dokler povezava dejansko ni vzpostavljena.

Kakor je bilo predstavljeno, ni tako. In to predvsem zavoljo uporabnikov samih – da bi bilo delo z brezžičnimi napravami le malo bolj priročno in udobno, kakor je že sedaj. Oddajanje okvirjev poskus zahteve bi lahko eliminirali z uporabo le pasivnega iskanja. Res smo tu odvisni predvsem od nastavitve intervalov usmerjevalnikov, v katerih naznanjajo svojo prisotnost v okolju, a je tudi ta pri večini proizvajalcev privzeto nastavljena na 100 milisekund, kar je zelo malo. Zakaj je torej dejansko v uporabi aktivno iskanje brezžičnih omrežij? Konkretnega odgovora na to vprašanje žal nismo našli, zato je to lahko bralcu v razmislek.

Predstavljeni so bili tudi primeri analiz zajetih podatkov – v eksperimentu smo videli, da se že v relativno kratkem času da izluščiti nekatere zanimive informacije. Če bi bil čas zajemanja daljši, na primer več let, bi verjetno lahko dobili informacije s še večjo dodano vrednostjo in, predvsem, natančnostjo.

Dotaknili smo se tudi pravnega vidika takšnega početja. Predvsem v današnjih časih, ko so mediji polni takšnih ali drugačnih razkritij, ko nekatere vladne organizacije dobesedno vohunijo za svojimi državljani na spletu, jim prisluškujejo, zbirajo ogromne količine podatkov iz socialnih omrežij, zahtevajo (omrežne) prometne podatke od večjih spletnih korporacij, in vse te podatke obdelujejo (pravijo da) z namenom zagotavljanja nacionalne varnosti [29] – takšno početje zagotovo ne bo sprejeto z odobravanjem. MAC naslov je enoličen – z nadaljnjim razširjanjem definicije osebnih (identifikacijskih) podatkov bi lahko nekoč tudi uradno postal priznan kot osebni podatek – povsod po svetu.

Namen diplomskega dela je bil navsezadnje tudi ta, da bralca opozori na problematiko zajemanja in obdelave (osebnih in drugih) podatkov in spodbudi k razmišljanju, kako to preprečiti oziroma vsaj zmanjšati.

Viri

[1] Brezžična omrežja. Dostopno 2013 na:

http://en.wikipedia.org/wiki/Wi-Fi

[2] Kratka zgodovina brezžičnih omrežij. Dostopno 2013 na:

http://www.arp.sprnet.org/default/inserv/trends/history_wireless.htm

[3] Brezžični standardi. Dostopno 2013 na:

http://en.wikipedia.org/wiki/IEEE_802.11

[4] Več o ISO/OSI referenčnem modelu. Dostopno 2013 na:

http://sl.wikipedia.org/wiki/ISO/OSI_referenčni_model

[5] Matthew Gast, 802.11 Wireless Networks: The Definitive Guide, O'Reilly Media, Inc. 2002. Dostopno 2013 na:

http://my.safaribooksonline.com/book/networking/wireless/0596001835

[6] Brezžični okvirji in tipi skeniranja. Dostopno 2013 na:

http://wirelessmscresearch.blogspot.com/2012/05/detecting-wardriving.html

[7] Usmerjevalnik TP-LINK 1043ND. Dostopno 2013 na:

http://www.tp-link.com/en/products/details/?model=TL-WR1043ND

[8] DD-WRT alternativna strojna programska oprema. Dostopno 2013 na:

http://www.dd-wrt.com/site/index

[9] SSH protokol. Dostopno 2013 na:

http://en.wikipedia.org/wiki/Secure_Shell

[10] GParted – programska oprema za upravljanje z diski. Dostopno 2013 na:

http://gparted.sourceforge.net/

[11] tcpdump – programska oprema za analizo omrežij. Dostopno 2013 na:

http://www.tcpdump.org/

[12] iwconfig – skripta za upravljanje brezžičnih vmesnikov v operacijskih sistemih Linux. Dostopno 2013 na:

http://www.linuxcommand.org/man_pages/iwconfig8.html

[13] Načini delovanja brezžičnih vmesnikov. Dostopno 2013 na:

http://wireless.kernel.org/en/users/Documentation/modes

[14] AWK – programski jezik za obdelavo nizov. Dostopno 2013 na:

http://www.grymoire.com/Unix/Awk.html

[15] XAMPP – popolno delujoč spletni strežnik. Dostopno 2013 na:

http://www.apachefriends.org/en/xampp.html

[16] JFreeChart – Java knjižnjica za generiranje diagramov. Dostopno 2013 na:

http://www.jfree.org/jfreechart/

[17] opkg – upravitelj paketov. Dostopno 2013 na:

http://wiki.openwrt.org/doc/techref/opkg

[18] Parametrizirane izjave. Dostopno 2013 na:

http://en.wikipedia.org/wiki/Prepared_statement

[19] Primeri JFreeChart diagramov. Dostopno 2013 na:

http://www.jfree.org/jfreechart/samples.html

[20] Več o času v Linux okolju. Dostopno 2013 na:

http://en.wikipedia.org/wiki/Unix_time

[21] Seznami MAC naslovnih prostorov po proizvajalcih. Dostopno 2013 na:

http://hwaddress.com/

[22] Več o MAC naslovih. Dostopno 2013 na:

http://en.wikipedia.org/wiki/MAC_address

[23] Hrošč v programu tcpdump. Dostopno 2013 na:

http://sourceforge.net/p/tcpdump/bugs/116

[24] Pogoji uporabe Spar plus kartice. Dostopno 2013 na:

http://www.spar.si/spar/spar_plus/splosnipogoji.htm

[25] Pogoji uporabe Mercator pika kartice. Dostopno 2013 na:

http://www.mercator.si/Static/upload/file/Splosna%20dolocila%20za%20izdajanje%20in%20uporabo%20placilno(5).pdf

[26] Shranjevanje zgodovine iskanja na Googlu in uporaba teh podatkov. Dostopno 2013 na:

https://support.google.com/accounts/answer/54053?hl=en

[27] PayPal pošilja na trg Beacon. Dostopno 2013 na:

http://techcrunch.com/2013/09/09/paypal-debuts-its-newest-hardware-beacon-a-bluetooth-le-enabled-device-for-hands-free-check-ins-and-payments/

[28] Bluetooh low energy. Dostopno 2013 na:

http://en.wikipedia.org/wiki/Bluetooth_low_energy

[29] ZDA – vladne agencije nadzirajo svoje državljane. Dostopno 2013 na:

http://www.theguardian.com/world/2013/aug/27/nsa-surveillance-program-illegal-aclu-lawsuit

[30] Kako vas trgovine sledijo s pomočjo vašega pametnega telefona. Dostopno 2013 na:

http://lifehacker.com/how-retail-stores-track-you-using-your-smartphone-and-827512308

[31] Mnenje informacijskega pooblaščenca RS o sledenju uporabnikov Bluetooth omrežja. Dostopno 2013 na:

https://www.ip-rs.si/varstvo-osebnih-podatkov/iskalnik-po-odlocbah-in-mnenjih/odlocbe-in-mnenja-varstvo-osebnih-podatkov/?tx_jzvopdecisions_pi1%5BshowUid%5D=2290&cHash=bdc8ec99e96442d6120edd2a09576507

[32] Program Wireshark. Dostopno 2013 na:

http://www.wireshark.org/

[33] Decibel milliwats. Dostopno 2013 na:

http://en.wikipedia.org/wiki/DBm

[34] Indoor position detection using wifi and trilateration technique, Nor Aida Mahiddin, Noaizan Safie, Elissa Nadia, Suhailan Safei, Engku Fadzli, Faculty of Informatics,University Sultan ZainalAbidin, Gong Badak Campus,Terengganu, Malaysia. Dostopno 2013 na:

http://sdiwc.net/digital-library/web-admin/upload-pdf/00000223.pdf